INTERNATIONAL STANDARD

ISO/IEC/IEEE 24641

First edition
2023-05

Systems and Software engineering — Methods and tools for model-based systems and software engineering

Ingénierie du logiciel et des systèmes — Méthodes et outils pour l'ingénierie du logiciel et des systèmes basée sur des modèles

Reference number
ISO/IEC/IEEE 24641:2023(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Institute of Electrical and Electronics Engineers, Inc
3 Park Avenue, New York
NY 10016-5997, USA
Email: stds.ipr@ieee.org
Website: www.ieee.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO/IEC documents should be noted. This document was drafted in accordance with the rules given in the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see https://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

ISO/IEC/IEEE 24641 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*, in cooperation with the IEEE Computer Society Systems and Software Engineering Standards Committee, under the Partner Standards Development Organization cooperation agreement between ISO and IEEE.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

As systems grow in scale and complexity, some in the systems engineering community turn to model-based systems and software engineering (MBSSE) to, among other objectives, manage complexity, maintain consistency, and help ensure traceability during system development. With an MBSSE approach, the systems and software engineering activities rely on evolving models that serve as the main or major source of knowledge about the system-of-interest and its life cycle processes, which could be any entity subject to a system model such as a program, project, product, or company.

MBSSE benefits differ significantly from 'engineering with models', which has been a common practice among the engineering disciplines for decades and that is mainly based on independent discipline-specific models that, even if very useful for each discipline and system analysis contribution, do not provide an overall understanding of the architecture of the system sharable among stakeholders, e.g. computer-aided design (CAD) for mechanical engineering, aerodynamics models, control loop simulations. In addition, due to the diversity of approaches and terminologies (e.g. model-driven development or MDD), MBSSE usually falls within the context of a specific engineering discipline (e.g. MDD for the software engineering community).

MBSSE is the formalized application of modelling to support systems engineering or software engineering activities. Faced with the issues and challenges linked to the growing complexity of the systems to be developed, document-centric approaches are less and less suitable. The MBSSE approach makes it possible to develop logically consistent multi-view architecture description. These serve as a bridge to enable the traceable, verifiable and dynamic correlation of the system-of-interest and/or software-of-interest models cross multidiscipline and throughout its entire life cycle, and to drive the system and software engineering processes, activities and tasks at all levels of its hierarchy from system-of-systems to system element across multiple engineering disciplines and throughout all stages of its life cycle

From MBSSE perspective, other engineering disciplines (mechanical, thermal, electronic, electrical, etc.) are also considered.

Thus, a need exists to specify the considerations necessary for undertaking the application of MBSSE within an organization. An organization needs to address the considerations necessary for supporting the establishment of each project environment within its overall ecosystem, and the exchange of models between stakeholder organizations.

This document addresses MBSSE-related processes by categorizing them into four process groups:

— Plan MBSSE

— Build models

— Perform MBSSE

— Support models

Each process is defined in terms of purpose, inputs, outcomes, and supporting tasks. The task descriptions include tool and method guidance and the recommended capabilities needed to successfully implement them. The relationships among the four process groups in this document, the four process groups in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207, and the life cycle model and stages in ISO/IEC/IEEE 24748-1 are described in Annex A.

This document is intended to benefit those who acquire, supply, develop, operate, and maintain MBSSE tools and methods. It can be used by:

a) organizations that need to implement or build models – to understand, adopt, and enact the MBSSE processes, tools, and methods (it also helps to evaluate and select relevant tools and methods based on business- and user-related criteria);

b) tool vendors who facilitate or leverage MBSSE practices – to provide a set of recommended tool capabilities for planning MBSSE, building models, MBSSE performance, and support.

Systems of systems are considered in this document to benefit from the same processes, methods and tool capabilities as any system.

The relationships between this document and other standards are described in Annex E.

NOTE 1 This document prescribes a way to engineer systems and software based on models thanks to a reference model and four process groups; however, other particular uses of models which are out of the scope of this document are used in "model engineering" in other ways: For example, in model-driven modernization [also called architecture-driven modernization (ADM) in object management group (OMG) terms], models are (automatically) generated from the existing code and artefacts of a running system in order to represent it and then build a new system in a different platform. Another usage scenario of models occurs in what is called "models@runtime" whereby the models are used to change the system and evolve with it; these are normally used in self-adaptive systems to achieve the required system self-adaptation features.

NOTE 2 The reference model does not take into account the system evolution (and that of its related models) as a fundamental phase of systems or software engineering in the maintenance and evolution of the system and its models.

NOTE 3 The design within the different domains, for example, mechanical, hydraulics, electrical, electronics, control algorithms, and software, has been performed using model-based techniques for decades. However, each domain uses specialized languages and tool chains for its modelling activities. The guideline to propose how the methods, modelling languages and tools apply in these domains is outside of the scope of this document. However, the interfaces of the engineering models and the system models are crucial and essential for applying MBSSE.

In this document, the following verbal forms are used:

— "shall" indicates a requirement;

— "should" indicates a recommendation;

— "may" indicates a permission;

Systems and Software engineering — Methods and tools for model-based systems and software engineering

1 Scope

This document deals with the tool capabilities and methods for model-based systems and software engineering (MBSSE). This document:

— specifies a reference model for the overall structure and processes of MBSSE-specific processes, and describes how the components of the reference model fit together;

— specifies interrelationships between the components of the reference model;

— specifies MBSSE-specific processes for model-based systems and software engineering; the processes are described in terms of purpose, inputs, outcomes and tasks;

— specifies methods to support the defined tasks of each process;

— specifies tool capabilities to automate or semi-automate tasks or methods.

This document does not bring any additional life cycle processes for system and software but specifies an MBSSE reference model considered as activities, not only from the life cycle perspectives of systems engineering problem solving and the system-of-interest evolution, but also from the cognitive perspectives of modelling and model management, which can sustain and facilitate the system and software life cycle processes during digital transformation and in the digital age.

The processes defined in this document are applicable for a single project, as well as for an organization performing multiple projects or an enterprise. These processes are applicable for managing and performing the systems and software engineering activities based on models within any stage in the life cycle of a system-of-interest.

2 Normative references

There are no normative references in this document.

3 Terms, definitions, and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO, IEC, and IEEE maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org

— IEEE Standards Dictionary Online: available at http://dictionary.ieee.org

NOTE For additional terms and definitions in the field of systems and software engineering, see ISO/IEC/IEEE 24765, which is published periodically as a “snapshot” of the SEVOCAB (Systems and Software Engineering Vocabulary) database and is publicly accessible at computer.org/sevocab.

3.1.1
analytical model
model (3.1.15) that describes mathematical relationships, such as differential equations that support quantifiable analysis about the *system* (3.1.35) parameters

Note 1 to entry: Analytical models can be further classified into dynamic and *static models* (3.1.34).

3.1.2
asset
item, thing, or entity that has potential, or actual value to an organization

Note 1 to entry: Assets can be classified from different perspectives, such as tangible assets, intangible assets; moveable assets, immoveable assets. Intangible assets can be classified into digital assets and non-digital intangible assets.

Note 2 to entry: Cognitive assets refer to intangible assets generated by an organization in the course of its operations. Data, information, knowledge, wisdom, and modelling assets are all belong to cognitive assets..

[SOURCE: ISO/IEC 19770-1:2017, 3.1, modified — The original three notes to entry have been replaced by two new notes to entry.]

3.1.3
capability
ability to do something useful under a particular set of conditions

Note 1 to entry: Generally, different kinds of capabilities exist: organizational capability, *system* (3.1.35) capability, and operational capability. Organizational capabilities relate through the work practices that are adopted by the organizations. New systems (with new or enhanced system capabilities) are developed to enhance enterprise operational capability in response to *stakeholders'* (3.1.32) *concerns* (3.1.5) about a problem situation. Operational capabilities provide operational services that are enabled by system capabilities. These system capabilities are inherent in the system that is conceived, developed, created, and/or operated by an enterprise. Enterprise SE concentrates its efforts on maximizing operational value for various stakeholders, some of whom can be interested in the improvement of some problem situation.

3.1.4
concept of operations
verbal and graphic statement, in broad outline, of an organization's assumptions or intent in regard to an operation or series of operations of new, modified, or existing organizational *systems* (3.1.35)

Note 1 to entry: The concept of operations frequently is embodied in long-range strategic plans and annual operational plans. In the latter case, the concept of operations in the plan covers a series of connected operations to be carried out simultaneously or in succession to achieve an organizational performance objective. See also *operational concept* (3.1.24).

Note 2 to entry: The concept of operations provides the basis for bounding the operating space, system capabilities, interfaces, and operating environment.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.9]

3.1.5
concern
matter of interest or importance to a *stakeholder* (3.1.32)

EXAMPLE Affordability, agility, availability, dependability, flexibility, maintainability, *reliability* (3.1.27), *resilience* (3.1.28), usability and viability are examples of concerns. Survivability, depletion, degradation, loss, obsolescence are examples of concerns. The PESTEL mnemonic is a reminder of possible areas of concern: political, economic, social, technological, environmental, and legal.

[SOURCE: ISO/IEC/IEEE 42020:2019, 3.8]

3.1.6
descriptive model
model (3.1.15) that shows an interconnected set of *model elements* (3.1.19) which represent key *system* (3.1.35) aspects including its structure, behaviour, parametric, and requirements

Note 1 to entry: Descriptive models are complementary to analysis and design models of a system.

3.1.7
discipline-specific model
representation of a *system* (3.1.35), or system elements from the perspective of a discipline addressing domain-specific *concerns* (3.1.5) where the *model elements* (3.1.19) come from a specific discipline

3.1.8
maturity
degree to which a *system* (3.1.35), product, or component meets needs for *reliability* (3.1.27) under normal operation

Note 1 to entry: The degree of maturity of a system, product or component can be associated with trust in, and good knowledge of, the behaviour of the system.

Note 2 to entry: The system, product or component is considered immature if there are still flaws or missing parts that prevent users from benefitting from the item. The item is considered mature if there are no flaws or missing parts that prevent users from benefitting from the item.

[SOURCE: ISO/IEC 25010:2011, 4.2.5.1, modified — The original note to entry has been replaced by two new notes to entry.]

3.1.9
maturity level
degree of achievement to which all goals have been attained

3.1.10
measure of effectiveness
MOE
operational measure of success that is closely related to the achievement of the operational objective being evaluated in the intended operational environment under a specified set of conditions

[SOURCE: ISO/IEC/IEEE 24748-4:2016, 4.7]

3.1.11
measure of performance
MOP
engineering parameter that provides critical performance requirements to satisfy a *measure of effectiveness (MOE)* (3.1.10)

Note 1 to entry: An MOP typically characterizes physical or functional attributes relating to the *system* (3.1.35) operation.

[SOURCE: ISO/IEC/IEEE 24748-4:2016, 4.8]

3.1.12
meta-model
special kind of model that specifies the abstract syntax of a modelling language

Note 1 to entry: The typical role of a meta-model is to define the semantics for how *model elements* (3.1.19) in a *model* (3.1.15) get instantiated. A model typically contains model elements. These are created by instantiating model elements from a meta-model (i.e. meta-model elements).

[SOURCE: ISO/IEC 19506:2012 Clause 4]

3.1.13
mission
important operational job or duty assigned to a *resource* (3.1.29) or a group of resources or certain groups of people

Note 1 to entry: A resource can be a human resource or a technical resource including *systems* (3.1.35) and products.

3.1.14
mode
definition of the expected behaviour of the *system* (3.1.35) (or of its actors, or of its components) in situations foreseen at design time

Note 1 to entry: Each mode is mainly characterized by the expected functional content of the system in this mode. A mode can reflect various concepts, such as:

— *phases* (3.1.25) of a *mission* (3.1.13) or of a flight for example (taxiing, taking-off, cruising, landing, etc.);

— specific required functioning of the system under certain conditions (connected, autonomous, etc.);

— specific conditions where the system is used; test, training, maintenance, etc.

The transition from one mode to another is in general the result of a decision, such as a change in the way the system operates, in order to adapt to new needs, or new contexts; it is therefore conditioned by the choices of the system, its users, or of external actors.

3.1.15
model
abstract representation of an entity or collection of entities that provides the ability to portray, understand or predict the properties or characteristics of the entity or collection under conditions or situations of interest

Note 1 to entry: A model can use a formalism that could be based on mathematical or scientific principles and concepts. A model can be generated using an established meta-model. Meta-models are often used to facilitate development of accurate, complete, consistent and understandable models.

Note 2 to entry: A model can be used to construct or express architecture views of the entity. *Descriptive models* (3.1.6) and analytic models are two kinds of models. A model should be governed by a model kind in accordance with ISO/IEC/IEEE 42010.

Note 3 to entry: A reference model can be used to capture a general case that is used as the basis for creating special case models for particular conditions or situations. A reference model can be used to encourage and enforce uniformity of architectures and architecture elements.

Note 4 to entry: The model can be an architecture model, architecture entity model, concept model or reference model, as the case may be.

Note 5 to entry: A physical model is a concrete representation that is distinguished from the mathematical and logical models, both of which are more abstract representations of the *system* (3.1.35). The abstract model can be further classified as descriptive (similar to logical) or analytical (similar to mathematical).

Note 6 to entry: (modelling and simulation) An approximation, representation, or idealization of selected aspects of the structure, behaviour, operation, or other characteristics of a real-world process, concept, or system. Models can have other models as components (Authoritative Dictionary of IEEE Standards Terms).

Note 7 to entry: An example of classification of models and relationships in MBSSE is given (see Annex C).

[SOURCE: ISO/IEC/IEEE 42020:2019, 3.13, modified — notes 5, 6 and 7 to entry have been added.]

3.1.16
model-based systems and software engineering
MBSSE
formalized applications of modelling to support *systems* (3.1.35) and software engineering

3.1.17
model baseline
immutable set of *model configuration items* (3.1.18) with their associated versions and variants

3.1.18
model configuration item
MCI
logical part of the *model* (3.1.15) that is maintained in a controlled fashion, having a trackable revision history

— Model (what is developed within a particular project)

— Each main package under the model root

— Catalogues which can be reusable as Libraries (e.g. Functions, Services, Measurements)

Note 1 to entry: A model as well as any *model element* (3.1.19) and its references can be part of a MCI. A MCI can be defined in different granularities, from a set of model elements, to the entire model. MCIs are managed to maintain the integrity of the models.

3.1.19
model element
atomic (elementary) item that represents an individual component, action, *state* (3.1.33), message, property, relationship, or another item that describes the composition, characteristics, or behaviour of a *system* (3.1.35)

3.1.20
model element library
set or catalogue of non-modifiable *model elements* (3.1.19) usable within any project, packaged in a single artefact

3.1.21
model pattern
general, reusable *model* (3.1.15) or model part that can be used as a solution to a commonly occurring problem within a given context in *system* (3.1.35) or software design

3.1.22
model repository
means to store different *models* (3.1.15) at different levels of abstraction and to facilitate understanding and cooperation between *stakeholders* (3.1.32) and practitioners at different levels

[SOURCE: Adapted from TOGAF 9.2 (The Open Group Architecture Framework)]

3.1.23
ontology
logical structure of the terms used to describe a domain of knowledge, including both the definitions of the applicable terms and their relationships

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.2691]

3.1.24
operational concept
verbal and graphic statement of an organization's assumptions or intent in regard to an operation or series of operations of a specific *system* (3.1.35) or a related set of specific new, existing or modified systems

Note 1 to entry: The operational concept is designed to give an overall picture of the operations using one or more specific systems, or set of related systems, in the organization’s operational environment from the users’ and operators’ perspectives. See also *concept of operations* (3.1.4).

Note 2 to entry: The operational concept is about systems, while a concept of operations typically refers to organizations.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.23]

3.1.25
phase
period of time in the life cycle during which activities are performed that enable achievement of objectives for that phase

[SOURCE: ISO/IEC/IEEE 42020:2019, 3.15]

3.1.26
reference framework
structure for understanding significant relationships among the entities of some environment, and for the development of consistent standards or specifications supporting that environment

Note 1 to entry: A reference framework provides a common backplane for consistency, collaboration, sharing, and reuse.

[SOURCE: ISO/IEC 20013:2020, 3.5, modified — The domain "<e-Portfolio>" has been removed; the original note to entry has been replaced by a new one.]

3.1.27
reliability
ability of the *system* (3.1.35) or the component to perform its required functions under stated conditions for a specified period of time.

3.1.28
resilience
ability of the *system* (3.1.35) to deliver required *capability* (3.1.3) in the face of adversity

3.1.29
resource
entity that is utilized or consumed during the execution of a process

EXAMPLE Diverse entities such as funding, personnel, facilities, capital equipment, tools, and utilities such as power, water, fuel and communication infrastructures.

[SOURCE: ISO/IEC/IEEE 12207:2017, modified — "asset" has been replaced by "entity"; note 1 to entry has been removed.]

3.1.30
safety
avoidance of injury or harm through the use or misuse of the *system* (3.1.35)

3.1.31
security
ability of the *system* (3.1.35) to withstand an attack, whether it is an intrusion, interference, or theft

3.1.32
stakeholder
role, position, individual or organization or classes thereof, having an interest, right, share, claim in an entity or its possession of characteristics that meets their needs and expectations

EXAMPLE End users, operators, acquirers, owners, suppliers, architects, developers, builders, maintainers, regulators, taxpayers, certifying agencies, and markets

Note 1 to entry: Some stakeholders can have interests that oppose each other, or oppose the *system* (3.1.35).

[SOURCE: ISO/IEC/IEEE 42020:2019, modified — "or classes thereof" has been added; "a right, share, claim or other interest" has been changed to "an interest, right, share, claim"; "architecture entity" and "architecture" have been replaced by "entity" and "possession of characteristics" respectively: "reflects" has been replaced by "meets"; EXAMPLE and note 1 to entry have been added.]

3.1.33
state
condition that characterizes a *system* (3.1.35), system element, function, or other entity at a point in time

[SOURCE: ISO/IEC/IEEE 29148:2018, 3.1.30, modified — "the behaviour of a function, subfunction or element" has been replaced by "a system, system element, function, or other entity"; note 1 to entry has been removed.]

3.1.34
static model
analytical model (3.1.1) of a *system* (3.1.35) in which there is no change

EXAMPLE A scale model of a bridge, studied for its appearance rather than for its performance under varying loads.

[SOURCE: The Authoritative Dictionary of IEEE Standards Terms]

3.1.35
system
arrangement of parts or elements that together exhibit a stated behaviour or meaning that the individual constituents do not

Note 1 to entry: A system is sometimes considered as a product or as the services it provides.

Note 2 to entry: In practice, the interpretation of its meaning is frequently clarified by the use of an associative noun, e.g. aircraft system. Alternatively, the word "system" is substituted simply by a context-dependent synonym (e.g. aircraft), though this potentially obscures a system principles perspective.

Note 3 to entry: A complete system includes all of the associated equipment, facilities, material, computer programs, firmware, technical documentation, services, and personnel required for operations and support to the degree necessary for self-sufficient use in its intended environment.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.46]

3.1.36
system element
discrete part of a *system* (3.1.35) that can be implemented to fulfil specified requirements

EXAMPLE Hardware, software, data, humans, processes (e.g. processes for providing service to users), procedures (e.g. operator instructions), facilities, materials, and naturally occurring entities, or any combination.

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.47]

3.1.37
system-of-interest
system (3.1.35) whose life cycle is under consideration

[SOURCE: ISO/IEC/IEEE 15288:2023, 3.48, modified — The abbreviated term "SoI" has been removed.]

3.1.38
technology map
outline of required and anticipated changes in technologies, with expected dates, which will enable achievement or transformation of a *system* (3.1.35)

[SOURCE: ISO/IEC 26560:2019, 3.6, modified — "product or product family" has been replaced by "system".]

3.1.39
use case
description of the behavioural requirements of a *system* (3.1.35) and its interaction with a user

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.15, modified — Note 1 to entry has been removed.]

3.1.40
viewpoint
set of conventions for the creation, interpretation and use of a *model* (3.1.15) view to frame one or more *concerns* (3.1.5)

[SOURCE: ISO/IEC/IEEE 42010:2022, 3.8, modified — The term has been changed from "architecture viewpoint" to "viewpoint"; "an architecture view" has been replaced by "a model view"; notes to entry have been removed.]

3.2 Abbreviated terms

BOM	bill of material
BPMN	business process model and notation
CI	configuration item
CIR	configuration item record
COTS	commercial-off-the-shelf
CPU	central processing unit
CRUD	create, read, update, and delete
CSA	configuration status accounting
CV	curriculum vitae
DBMS	database management systems
DIKW	data information knowledge wisdom
FAST	function analysis system technique
FFBD	functional flow block diagram
FMEA	failure mode and effects analysis
FTA	fault tree analysis
HFE	human factors engineering
HLA	high level architecture
ICD	interface control document
ILS	integrated logistics support
IPR	intellectual property rights
IV and V	integration verification and validation
KSA	knowledge, skills and abilities
MBSA	model-based safety assessment
MBSSE	model-based systems and software engineering
MCI	model configuration item

MDD	model-driven design
MOE	measure of effectiveness
MOP	measure of performance
OOSEM	object oriented systems engineering method
OPM	object-process methodology
PBS	product breakdown structure
RAMT	reliability availability maintainability testability
SBS	system breakdown structure
SE	systems engineering
SSE	systems and software engineering
SysML	systems modelling language
UML	unified modelling language

4 Conformance

4.1 Intended usage

The requirements in this document are contained in Clauses 5, 6, 7, 8 and 9. This document provides requirements for a number of processes, tasks, methods and tool capabilities for MBSSE suitable for usage during the system and software engineering processes, activities and tasks at all levels of its hierarchy from system-of-systems to system element and throughout all stages of its life cycle. It is recognized that particular projects or organizations may not need to use all of the processes provided by this document. Therefore, implementation of this document typically involves selecting and declaring a set of processes suitable to the organization or project. There are two ways that an implementation can be claimed to conform to the provisions of this document – full conformance and tailored conformance.

There are two criteria for claiming full conformance. Achieving either criterion suffices for conformance, although the chosen criterion (or criteria) shall be stated in the claim. Claiming “full conformance to tasks” asserts that all of the requirements of the tasks, methods and tool capabilities of the declared set of processes are achieved. Alternatively, claiming “full conformance to outcomes” asserts that all of the required outcomes of the declared set of processes are achieved. Full conformance to outcomes permits greater freedom in the implementation of conforming processes and can be useful for implementing processes to be used in the context of an innovative life cycle model.

NOTE 1 Options for conformance are provided for needed flexibility in the application of this document. Each process has a set of objectives (phrased as “outcomes”) and a set of tasks, methods and tool capabilities that represent one way to achieve the objectives.

NOTE 2 Users who implement the tasks, methods and tool capabilities of the declared set of processes can assert full conformance to tasks of the selected processes. Some users, however, can have innovative process variants that achieve the objectives (i.e. the outcomes) of the declared set of processes without implementing all of the tasks, methods and tool capabilities. These users can assert full conformance to the outcomes of the declared set of processes. The two criteria – conformance to task and conformance to outcome – are necessarily not equivalent since specific performance of tasks, methods and tool capabilities can require, in some cases, a higher level of capability than just the achievement of outcomes.

NOTE 3 When this document is used to help develop an agreement between an acquirer and a supplier, clauses of this document can be selected for incorporation in the agreement with or without modification. In this case, it is more appropriate for the acquirer and supplier to claim compliance with the agreement than conformance with this document.

4.2 Full conformance

4.2.1 Full conformance to outcomes

A claim of full conformance declares the set of processes for which conformance is claimed. Full conformance to outcomes is achieved by demonstrating that all of the outcomes of the declared set of processes have been achieved. In this situation, the provisions for tasks, methods and tool capabilities of the declared set of processes are guidance rather than requirements, regardless of the verb form that is used in the provision.

One intended use of this document is to facilitate process assessment and improvement. For this purpose, the objectives of each process are written in the form of 'outcomes' compatible with the provisions of ISO/IEC 33002. That standard provides for the assessment of the processes of this document, providing a basis for improvement. Users intending process assessment and improvement may use the process outcomes written in this document as the "process reference model" required by ISO/IEC 33002.

4.2.2 Full conformance to tasks

A claim of full conformance declares the set of processes for which conformance is claimed. Full conformance to tasks is achieved by demonstrating that all of the requirements of the tasks, methods and tool capabilities of the declared set of processes have been achieved. In this situation, the provisions for the outcomes of the declared set of processes are guidance rather than requirements, regardless of the verb form that is used in the provision.

NOTE A claim of full conformance to tasks can be appropriate in contractual situations where an acquirer or a regulator requires detailed understanding of the suppliers' processes.

4.3 Tailored conformance

When this document is used as a basis for establishing a set of processes that do not qualify for full conformance, the clauses of this document are selected or modified. The tailored text, for which tailored conformance is claimed, is declared. Tailored conformance is achieved by demonstrating that the outcomes, tasks, methods and tool capabilities, as tailored, have been achieved.

5 MBSSE reference model

5.1 Overview

The MBSSE reference model, as shown in Figure 1, specifies MBSSE processes structured into four process groups:

— plan MBSSE;

— build Models;

— perform MBSSE;

— support Models.

The rest of this document describes tasks, methods and tools in terms of the process groups and MBSSE-specific processes defined in the reference model. Annex D describes the key roles of model-based systems and software engineering.

The MBSSE reference model divides each process group into MBSSE-specific process and describes each process in terms of the following attributes:

— name of the process;

— purpose of the process;

— inputs to produce outcomes;

— process outcomes;

— tasks to achieve the outcomes;

— method and tool capabilities for performing tasks effectively and efficiently.

Method capability shall be understood as an ability to implement tasks in a systematic or established fashion.

The method in this document:

— reduces time and effort when conducting impact analyses and trade studies;

— implements administrative and support task that is quite different from modelling methods.

NOTE See ISO/IEC/IEEE 24774 for further guidance on specification for process description.

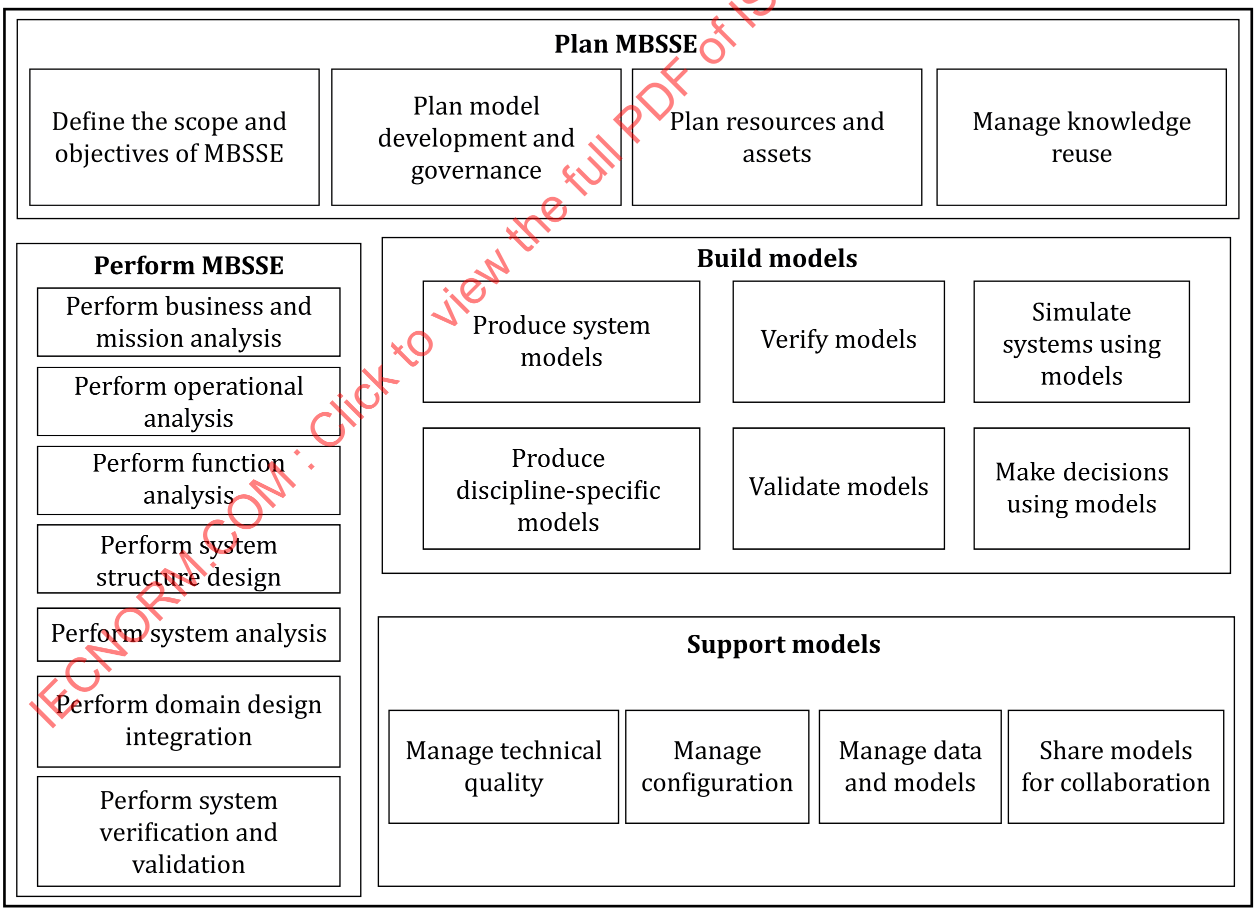

Figure 1 — MBSSE Reference model

The first group, "Plan MBSSE" identifies processes required as prerequisites for modelling. Any MBSSE deployment shall go through these processes. "Plan MBSSE" processes are performed at the beginning of the systems and software modelling life cycle. The scope and objectives help select systems and software engineering concerns to be addressed.

The “Plan MBSSE” process group identifies processes dedicated to the model development organization that may be considered necessary for each specific modelling project or either independent of any specific modelling effort (e.g. plan model development and governance, or plan resources and assets) and in this case, the associated tasks are non-recurring, or not necessary for each project.

The other two main process groups, “Build models” and “Support models” apply to any specific MBSSE process defined in the “Perform MBSSE” group. “Build Models” processes are core-modelling processes and help ensure consistency in the model repository. “Support models” processes deal with technical data and are performed during the whole life cycle.

The “Perform MBSSE” processes included in the MBSSE management plan should include the systems engineering (SE) concerns identified during the “Plan MBSSE” processes. These processes specify which kind of models are to be produced and use all the other processes described within three main process groups.

This MBSSE reference model aims to support iterative workflows, enabling several iterations between the processes of different groups, to progressively reach adequate MBSSE definition according to defined objectives and strategy.

The relationships among these process groups are presented by Figure 2.

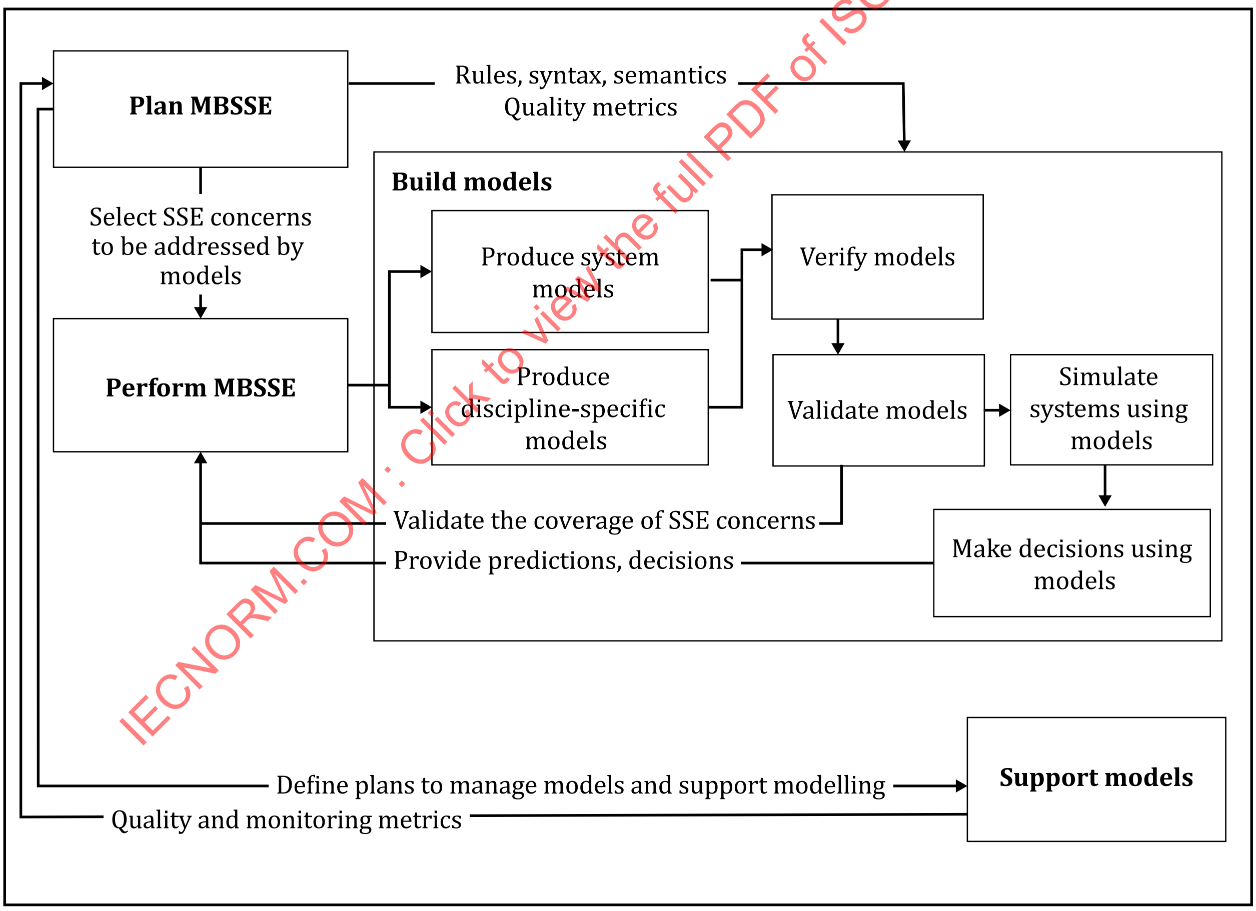

Figure 2 — The relationships among MBSSE process groups and build models processes

5.2 Build models processes and data-information-knowledge-wisdom (DIKW)

The primary objective is building useful models in an effective and collaborative way. The core modelling activities can be put in perspective through the lenses of the data-information-knowledge-wisdom (DIKW) (See Reference [39]) pyramid:

— producing models makes system engineering data useful by transforming them to shareable information and knowledge;

— verifying and validating models increases the confidence and the accuracy of the knowledge gained;

— analysing and simulating systems using models allows us to predict the future correctly, not only by detecting and understanding patterns but also deeply comprehending the "why" behind those patterns.

The relationships among the build models processes and the data-information-knowledge-wisdom pyramid are presented by the Figure 3;

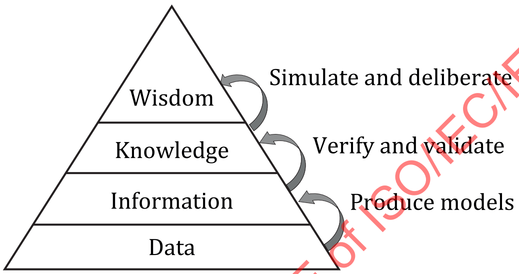

Figure 3 — Build models processes and data-information-knowledge-wisdom (DIKW) pyramid

Figure A.2 provides an example of a cognition dimension based on the DIKW pyramid.

6 Plan MBSSE

6.1 General

The plan MBSSE process group gathers MBSSE-enabling prerequisites, contributing to the effectiveness and efficiency of the approach both within the project and at the enterprise level. It encompasses assessing and scoping, planning, and technical management (setting up the MBSSE approach) along with resources, knowledge, and knowledge management activities (for the mid-term and long-term perspectives).

As the MBSSE management plan has to be provided, for practical reasons, it may be included and described in the systems engineering management plan instead of producing a separate MBSSE management plan.

Plan MBSSE includes the following processes and the tools and methods that support them:

— define the scope and objectives of MBSSE;

— plan model development and governance;

— plan resources and assets;

— manage knowledge reuse.

6.2 Define the scope and objectives of MBSSE

6.2.1 Principal constituents

6.2.1.1 Purpose

This process defines the scope and objectives of MBSSE and is expected to increase the value added by performing MBSSE. Tasks included in this process identify system or software engineering concerns to be addressed and the depth and breadth of models to be produced.

NOTE 1 It is important to decide on the objectives and on the scope of modelling: the granularity, where to stop, the interoperability constraints, the synchronization with the other MBSSE models or other MBSA and the expected sequences of (re)synchronization.

NOTE 2 The MBSA approach is defined as abstracting a physical model into a formal model consisting of the failure behaviour of the system and its' components. The main physical model can be extended or an exclusive failure model can be used.

6.2.1.2 Inputs

The following inputs should be available to perform the 'Define the scope and objectives of MBSSE' process:

a) stakeholder needs, requirements, concerns and expectations, and expected model usage;

b) models of interest; that will need to be created and that already exist such as the model library(ies);

c) system and software requirements related to specific topics (e.g. functional, non-functional, integration, development, production, procurement, costs, delays, logistics, knowledge, skills and abilities (KSA), etc.) into which the MBSSE helps gain insight;

d) quality and modelling life cycle monitoring objectives (expected measures and indicators, expected periodicity);

e) overall strategic goals of the entity (program, project, product, company);

f) business data model (meta model) to assess the candidate modelling methods and tools;

g) available MBSSE methodologies; some parts of a methodology can be used as applicable to the MBSSE context and expectations;

h) available modelling languages, with their applicable domains;

i) available modelling tools for supporting:

  1) MBSSE methodology and modelling conventions;

  2) interoperability with other engineering tools (e.g. for requirements management, traceability, and so on) and environments.

6.2.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the 'define the scope and objective of MBSSE' process.

a) Knowledge on the context of the system-of-interest, and current MBSSE capability of the project realizing system (e.g. stakeholders KSA for model understanding, production and review) is characterized.

b) MBSSE objectives and strategies are established.

c) Domain ontologies, required perspectives, required models and views and their levels are defined.

d) Quality and modelling life cycle monitoring objectives and strategies are established.

NOTE Knowledge can be understood (including but not limited to) as a set of quantifiable items such as constraints list, capability descriptions, data fidelity descriptions, model metrics, etc.

6.2.1.4 Tasks

The organization shall implement the following tasks with respect to the 'define the scope and objectives of MBSSE' process.

— Establish MBSSE goals and measures for the successful achievement of the entity level (program, project, product, company) goals.

— Specify the key elements of the MBSSE approach to define the most suitable MBSSE enablers to achieve the defined goals.

6.2.2 Establish MBSSE goals and measures

The goal of this task is to define and measure the MBSSE goals and strategies contributing to the successful achievement of the goals of a program or project or the organization.

a) A method supporting this task should contain the following capabilities:

   1) assessing the current MBSSE capability within the entity (program, project, product, company): this assessment has an effect in some MBSSE goals;

   2) defining collaboratively the MBSSE Objectives, progress measures, strategies, key drivers and perspectives;

   3) defining the target usages and types of users or consumers (i.e. knowledge sharing, communication, semi-formal or formal description, artefacts generation, etc.).

b) Management tool capabilities should support establishing MBSSE management goals by allowing the user to do the following:

   1) assess and report current MBSSE capability;

   2) access MBSSE goals and the program or project or organizational level plans;

   3) communicate the MBSSE management goals and strategies with key stakeholders by supporting channels and implemented mechanisms.

6.2.3 Specify the key elements of the MBSSE approach

This task defines the key elements of the MBSSE approach adapted to the context and goals.

a) The method should provide the following capabilities to define the key elements:

   1) establishing and maintaining the MBSSE concepts and ontologies;

   2) defining the languages (e.g. usage of domain-specific languages or universal languages);

   3) defining the usage of meta-models (e.g. fixed or evolving);

   4) identifying and analysing the risks mitigation means;

   5) establishing that the model structure conforms to an appropriate ontology according to the target usages and users; model structure is defined in terms of parts such as sections, perspectives, layers, abstraction layers, sub-models;

   6) setting the expected level of representativeness of each model;

   7) defining model integration, verification and consistency strategies;

   8) identifying the different analyses and simulations that are to be conducted with those models;

   9) defining the key drivers for MBSSE to tailor or define the detailed activities.

b) Tool capabilities should support defining key elements for MBSSE goals by allowing the user to do the following:

   1) share the MBSSE concepts, ontologies, languages, meta-models with the relevant organization units and participants;

   2) access MBSSE key element related artefacts produced during MBSSE governance;

   3) share risk sources;

   4) accumulate historical data for capturing risks;

   5) identify potential risks and opportunities associated with key elements.

6.3 Plan model development and governance

6.3.1 Principal constituents

6.3.1.1 Purpose

The purpose of this process is to develop an MBSSE model strategy, schedules and procedures for model building, model reviewing, model use, MBSSE risk mitigation, MBSSE benefit tracking and MBSSE governance to achieve the intended benefits.

The model strategy identifies the modelling need according to engineering defined goals: which models, model layers, viewpoints, model interactions are necessary for who.

NOTE The outcomes of this process can be captured in a development or management plan for MBSSE.

6.3.1.2 Inputs

The following inputs should be available to the 'plan model development and governance' process:

a) context, goals and measures as well as key elements of the MBSSE approach;

b) stakeholders expectations, needs, requirements and concerns of the modelling life cycle;

c) risk tolerance for stakeholders, which influences the risk mitigation efforts;

d) quality and life cycle monitoring objectives.

6.3.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the 'plan model development and governance' process.

a) Model development objectives and strategy are defined.

b) Cost estimates associated with the model development and governance activities.

c) MBSSE activities, risks, inventory, and associated analysis and mitigation means (e.g. KSA, tooling, over or under modelling) are defined.

d) MBSSE risks, issues and changes management and corrective process definition are defined.

e) Model life cycles are defined.

f) A model development schedule tied to program development milestones is defined.

g) Reviews roadmap is defined.

h) MBSSE guidance is available:

   1) methodological - describing the methodology to be applied and providing guidance and rules;

      NOTE 1 This methodological guide can also state the justifications of the methodology choice, and of its potential customization.

   2) modelling -describing the formalism to be applied, modelling guidance and rules;

      NOTE 2 This modelling guide can also state the justifications of the formalism choice, and of its potential customization or profiling.

   3) tooling - identifying the modelling tool supporting the defined MBSSE methodology and formalism.

i) Implementations of model verification (i.e. the correctness and completeness of model content), for the target tools and formalisms are established.

j) Implementations of expected model conformance indicators, metrics and success criteria evaluation means, against identified methodological rules, modelling rules and patterns are established.

k) Implementations of expected MBSSE activities progress indicators and metrics are established.

l) MBSSE trainings to disseminate methodology, modelling guidance and tool usage within the teams are managed.

m) Required performance/capability of model development and governance is defined.

n) Improvement either on the established model development and governance goals or on its achievement is performed.

NOTE As an entity evolves over time to include or eliminate common infrastructure services, leased services or other capabilities and functions, the MBSSE planning processes accommodate periodic changes to a management plan.

6.3.1.4 Tasks

The organization shall implement the following tasks with respect to the 'plan model development and governance' process.

— Define the MBSSE deployment procedure.

— Define the MBSSE life cycle flow; this task defines a high-level vision of the life cycle of the models and the development strategy of these models.

— Define the MBSSE methodology to be applied or tailored as appropriate with justification.

— Specify how to manage and control the modelling life cycle process, which identifies key elements, procedures, tools and parameters for assessing the effectiveness of the model management activities and its results. Improvements are identified and their possibilities are validated regarding the management of models activities and its results.

— Document the MBSSE management plan; this task creates and shares the plans.

— Improve model development and governance process continuously.

6.3.2 Define MBSSE deployment procedure

The goal of this task is to tailor and define MBSSE deployment procedures and to share the defined procedures with key stakeholders.

a) The method should support this task with the following capabilities:

1) defining MBSSE activities planning (including allocation to resources);

2) identifying the collaborative engineering interactions with other potential engineering phase, other concurrent engineering activities (e.g. requirements management, change management, version management, integration verification validation management, risk management, safety, security), previous and next engineering phase;

3) maintaining the consistency of the interfaces with other development tools (e.g. specific engineering disciplines tools);

4) specifying the constraints due to the runtime platform(s), access and control of access, replication and synchronization, performance (i.e. response times), availability, and backup/recovery; see 8.4.3;

5) maintaining the consistency of the MBSSE management plan with other potential engineering discipline plans, and systems engineering management plans to facilitate collaborative engineering;

6) defining monitoring, measuring, and controlling the effectiveness of the MBSSE activities and the associated risks management.

b) Tool capabilities should support defining key procedures for MBSSE management goals by allowing the user to do the following:

1) access the existing procedures for MBSSE management;

2) share key procedures with relevant key stakeholders;

3) integrate and calibrate procedures;

4) document the defined procedures electronically.

6.3.3 Define the MBSSE life cycle flow

The goal of this task is to define a high-level vision of the life cycle of the models and the development strategy of these models.

This strategic planning:

— should be detailed along the MBSSE life cycle flow;

— serves as an input to the choice of MBSSE methods and formalisms.

a) The method should support the definition of the MBSSE life cycle flow goals with the following capabilities:

1) defining which models are produced at each stage or phase, and how models are enhanced and updated after creation in successive states and phases;

2) managing the dependencies between the models produced at different stages of the development;

3) defining the required maturity of models at each stage of development: the information they are required to carry and the information not required at that time;

   4) defining the competency profiles of the resources needed (human, enablers) to bring the models to the required maturity stage;
   5) defining the training, or other development opportunities, to raise the modelling KSA to the required level;
   6) defining the maturity milestones of the models;
   7) defining the model acquisition milestones (when developed by suppliers).

b) Tool capabilities should support documenting the MBSSE life cycle flow goals by allowing the user to do the following:
   1) access a vision of the life cycle of the models and the development strategy;
   2) communicate the high-level vision and goals with key stakeholders;
   3) share the MBSSE life cycle status and models maturity level with relevant participants.

6.3.4 Define the MBSSE methodology

The goal of this task is to define or tailor the MBSSE methodology.

The MBSSE methodology includes the formalisms to be used for modelling and the tool capabilities.

a) The method should support defining the MBSSE methodology with the following capabilities:
   1) defining the key concepts, the traceability between key concepts, the process for modelling those key concepts in order to accomplish stakeholder objectives, and the inputs and outputs of each process;
   2) defining the MBSSE methodology, preferably from a set of available candidate methodologies, potentially to be tailored according to the MBSSE context;
   3) defining the MBSSE formalism from a set of available candidate formalisms, potentially to be customized or profiled according to the identified methodology and the MBSSE context;
   4) documenting and sharing the MBSSE methodological and modelling guides;
   5) preparing and delivering the associated training.

b) Tool capabilities should support documenting the MBSSE methodology goals by allowing the user to do the following:
   1) access defined MBSSE methodology and its goals;
   2) allow customization or profiling according to the MBSSE methodology;
   3) help ensure the accuracy and configuration of the information;
   4) verify the model meets rules for that diagram type;
   5) execute models to ensure they have no logic errors.

6.3.5 Specify how to manage and control the modelling life cycle process

The goal of this task is to prepare means for managing and controlling the modelling life cycle process.

a) The method should support specifying how to manage and control the modelling life cycle process goals with the relevant following capabilities:
   1) defining the MBSSE reviews roadmap;

2) defining the quality and life cycle monitoring elements (i.e. indicators, dashboards, etc.) according to the MBSSE objective;

3) configuring the potential monitoring and reporting tools of the MBSSE model development environment accordingly;

4) configuring the potential change management tools of the MBSSE model development environment, according to the MBSSE domain specificities (i.e. models artefacts types, defect types, etc.);

5) implementing for the target tools and formalisms the model verification and consistency checking rules as specified in the MBSSE modelling guide;

6) implementing the extraction or collection procedures for the expected model conformance and indicators, metrics and success criteria;

7) implementing the extraction or collection procedures for the expected MBSSE activities progress indicators and metrics;

8) institutionalizing sizing parameters and related efforts for the MBSSE activities;

9) proposing potential MBSSE process improvements;

10) periodically producing the quality and life cycle monitoring elements as defined in the MBSSE management plan.

b) Tool capabilities should support specifying how to manage and control the modelling life cycle process goals by allowing the user to do the following:

1) manage the configuration of the information provided by the models and other data elements (see Clause 8 for more details);

2) access historical data related to monitoring and controlling other processes;

3) specify key elements, procedures, tools and parameters using document standards.

6.3.6 Document the MBSSE management plan

The goal of this task is to document the MBSSE management plan, which is initiated during the 'defining scope and objectives of MBSSE' process.

a) The method should support documenting the MBSSE management plan goals with the following capabilities:

1) documenting and sharing the MBSSE management plan;

2) identifying and managing the model development risks and issues;

3) planning model development significant development achievements (e.g. schedule and milestones) for progress monitoring.

b) Tool capabilities should support documenting the MBSSE management plan goals by allowing the user to do the following:

1) write the MBSSE management plan or generate from the repository;

2) review and approve the MBSSE management plan.

6.3.7 Improve model development and governance process continuously

The goal of this task is to examine gaps between deployed and planned, and to provide the improved enablers.

a) The method should support improving model development and governance process continuously with the following capabilities:

   1) collecting data for evaluating the effectiveness of model development and governance;

   2) analysing deviations from required performance/capability of model development and governance necessary to achieve the established model development and governance goals;

   3) establishing action plans and success measures on improvement activities for achieving model development and governance goals;

   4) controlling and tracing the status of improvement activities to closure.

b) Tool capabilities should support improving model development and governance process continuously by allowing the user to do the following:

   1) accumulate data related to the improvement of model development and governance process;

   2) visualize deviation between the actual and expected effectiveness of model development and governance;

   3) share improvement activities with relevant participants through the communication channels and implemented mechanisms;

   4) check the status of improvement activities.

6.4 Plan resources and assets

6.4.1 Principal constituents

6.4.1.1 Purpose

The purpose of this process is to plan resources and assets, including required tool chains and infrastructures, for performing MBSSE effectively and efficiently. Plans vary according to the selected MBSSE approach.

6.4.1.2 Inputs

The following inputs should be available to 'plan resources and assets' process:

a) MBSSE management plan stating the MBSSE context;

b) MBSSE methodological and modelling guides;

c) knowledge, skills and abilities (KSA) necessary to apply the identified methodology and perform the modelling activities;

d) available competency framework standard as a mean of assessing KSA;

e) available personnel (i.e. curriculum vitae (CV), motivations) and their competency level in the required MBSSE methodology, modelling formalisms and tools;

f) types of models to be developed at each stage of the MBSSE development effort;

g) product Lines managed assets;

h) instances of Product Lines managed assets;

i) catalogues containing modelling patterns, potentially applicable to produce all or part of the expected MBSSE models;

j) similar legacy models, potentially reusable or customizable to produce the expected MBSSE models.

6.4.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the 'plan resources and assets' process.

a) Roles, KSA, responsibilities and associated activities are defined.

b) Identified personnel, and their required training or development opportunities, is known.

c) The set of modelling assets according to the MBSSE context is identified:

   1) any model asset customizations, product lines;

   2) legacy models from the available product baseline, catalogues and legacy models;

   3) product's model instances, patterns, patterns evolution proposal.

6.4.1.4 Tasks

The organization shall implement the following tasks with respect to the 'plan resources and assets' process.

— Define the MBSSE roles, KSA and associated responsibilities, activities or tasks and privileges, regarding MBSSE.

— Identify competent MBSSE skilled resources to perform the identified roles, and perform the identified activities or tasks.

— Identify and manage modelling assets enabling improvement of modelling quality and productivity.

6.4.2 Define the MBSSE roles, responsibilities, knowledge, skills and abilities (KSA)

The goal of this task is to describe an approach for determining what roles, responsibilities and KSA may be needed to accomplish the modelling scope.

a) The method should support the definition of MBSSE roles, responsibilities, and KSA goals with the following capabilities:

   1) defining the stakeholders with respect to the modelling objectives;

   2) considering what different KSA are necessary at different phases of the life cycle of the model, not limited to simply modelling, but also integrating the model with the rest of the technical and programmatic efforts;

   3) confirming planned training to promote the development of necessary KSA;

   4) defining the MBSSE roles, associated responsibilities and KSA.

   NOTE Typical examples are proposed in Table D.1.

b) Tool capabilities should support the definition of MBSSE roles, responsibilities and KSA goals by allowing the user to do the following:

   1) establish a stakeholder map with the associated expected KSA;

   2) plan the development of necessary KSA;

   3) generate a matrix for allocation of roles vs KSA.

6.4.3 Identify resources

The goal of this task is to identify resources.

a) The method should support the identification of resources goals with the following capabilities:

   1) searching for competent skilled resources able to perform the identified roles, support the requested responsibilities and perform the tasks;

   2) assessing or checking the KSA level;

   3) identifying the required training.

b) Tool capabilities should support the identification of resource goals by allowing the user to do the following:

   1) generate a matrix of expected KSA vs training required.

6.4.4 Manage modelling assets

The goal of this task is to manage modelling assets.

a) The method should support manage modelling assets goals with the following capabilities:

   1) searching for applicable product line assets, patterns and or legacy models according to the MBSSE context.

b) Tool capabilities should support manage modelling assets goals by allowing the user to do the following:

   1) access product line assets, patterns and or legacy models.

6.5 Manage knowledge reuse

6.5.1 Principal constituents

6.5.1.1 Purpose

The purpose of this process is to prepare and help optimize modelling efforts by reusing existing models and reference architectures, this also defines how to make models to be built in a reusable way and how to reuse the knowledge about or from elements: meta models, patterns, models, methods and tools.

6.5.1.2 Inputs

The following inputs should be available to the 'manage knowledge reuse' process:

a) modelling assumptions and scope;

b) acquired knowledge on the maturity of the model - i.e. the trust that can be placed in it, which have enabled it to be validated (which analytical models or tests, etc.) its representativeness, any problems encountered / limitations of representativeness;

c) MBSSE management plan and particularly the key drivers, against which trade-offs have to be made;

d) risks potentially identified in the MBSSE management plan to be considered in trade-off analysis;

e) models, potentially integrating multiple contributions from various stakeholders, and potentially containing consistency issues, and rule violations;

f) reference architecture models;

g) modelling tools supporting model integration or assembly, comparisons and merges (i.e. iterative and conservative import or export, synchronizations and diff-merge features).

6.5.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the ‘manage knowledge reuse’ process.

a) Modelling assumptions and scope are updated.

b) Variability and commonality analysis results for models coming from various sources are performed.

c) Model element library with support to search models is defined.

d) Models updated with the best results of model integration and trade-off analysis, restoring models consistency against modelling rules and preserving key-drivers are checked and available.

e) Trade-off decision justifications are established.

f) Model patterns and meta-models defined for patterns are identified.

g) Models are stored and managed in the repository according the taxonomy.

h) Knowledge reuse on methods is managed.

i) Knowledge reuse on tool extensions is managed.

6.5.1.4 Tasks

The organization shall implement the following tasks with respect to the ‘manage knowledge reuse’ process.

— Identify model patterns and define meta-models for patterns.

— Perform commonality and variability analysis.

— Manage the model repository.

— Manage knowledge reuse on methods.

— Manage knowledge reuse on tool extensions.

6.5.2 Identify model patterns and define meta-models for patterns

The goal of this task is to identify common model patterns in order to facilitate knowledge reuse on models.

a) The method should support the goal to identify model patterns and define meta-models for patterns with the following capabilities:

   1) defining a taxonomy of model patterns;

   2) collecting models from different sources including reference architecture models;

   3) identifying patterns;

    4) mapping to existing meta-models or define new meta-models corresponding to the patterns.

b) Tool capabilities should support the goal to identify model patterns and define meta-models for patterns by allowing the user to do the following:

    1) facilitate the taxonomy creation;

    2) assist the user in performing multiple model integration;

    3) support in pattern and meta-model creation and storage.

6.5.3 Perform commonality and variability analysis

The goal of this task is to extract common models from different sources in order to prepare for reuse.

a) The method should support the goal to perform commonality and variability analysis with the following capabilities:

    1) collecting models from different sources;

    2) identifying common patterns and variations;

    3) extracting models conforming to the patterns and store them;

    4) refactoring models to adapt to the patterns.

b) Tool capabilities should support the goal to perform commonality and variability analysis goals by allowing the user to do the following:

    1) classify models coming from different sources according to the taxonomy and store them;

    2) extract models, refactor and store them.

6.5.4 Manage the model repository

The goal of this task is to store the extracted and adapted models and manage the model repository in order to prepare for reuse.

a) The method should support the goal to manage the model repository with the following capabilities:

    1) classifying and storing models according to the taxonomy;

    2) defining keywords;

    3) providing a model search capability;

    4) managing changes and versions.

b) Tool capabilities should support the goal to manage the model repository goals by allowing the user to do the following:

    1) store models;

    2) attribute keywords to models;

    3) provide a portal to facilitate model search and visualization;

    4) extract models and reuse;

    5) manage changes and versions.

6.5.5 Manage knowledge reuse on methods

The goal of this task is to manage knowledge reuse on methods tailored for a particular context in order to reuse them within other similar contexts.

a) The method should support the goal to manage knowledge reuse on methods with the following capabilities:

   1) investigating the method used within projects or programs in regular basis;

   2) managing the changes and versions;

   3) writing method guides;

   4) making available for reuse and provide required training.

b) Tool capabilities should support the manage knowledge reuse on methods by allowing the user to do the following:

   1) provide a portal with information on methods and contexts;

   2) manage change and versions.

6.5.6 Manage knowledge reuse on tool extensions

The goal of this task is to manage knowledge reuse on tool extensions developed for a particular context in order to reuse them within other similar contexts.

NOTE Tool extensions can be understood (including, but not limited to) as a set of items such as plug-ins, scripts.

a) The method should support the goal to manage knowledge reuse on tool extensions with the following capabilities:

   1) providing a portal with information on tools and contexts;

   2) managing tools and user guides.

b) Tool capabilities should support the manage knowledge reuse on tool extensions by allowing the user to do the following:

   1) provide a portal with information on tool extensions;

   2) manage tool extensions and user guides.

7 Build models

7.1 General

These are the core modelling processes in an MBSSE approach, framed, monitored and supported by the contributing activities identified in the other groups in the MBSSE reference model.

The following processes fall within the build models process group:

— produce system models;

— produce discipline-specific models;

— verify models;

— validate models;

— simulate systems using models;

— evaluate alternative models.

7.2 Produce system models

7.2.1 Principal constituents

7.2.1.1 Purpose

The purpose of this process is to produce systems models, either integral to the conduct of requirement and design engineering by domain experts, or from engineering data collected from domain experts.

The system model can be understood as a model targeting the system features (or characteristics) of a entity of interest. In system context, these models should be multidisciplinary (or interdisciplinary or non-disciplinary oriented), meaning not targeting at a discipline in particular, but allowing multidisciplinary stakeholders to build, understand and share the model (see Annex B).

System models can include a combination of geometric, quantitative, and logical models. They often span several modelling domains such as different systems (e.g. thermal, power), different technology domains (e.g. hardware, software), and different characteristics (e.g. physical, performance). Each of these models shall be integrated to ensure a consistent and cohesive system representation. As such, the system model shall enable representation of general-purpose system modelling concepts such as behaviour and structure that can be shared across modelling domains.

Some examples of system models include the following (from ISO/IEC/IEEE 15288):

— a functional model that captures the system functions and their functional interfaces;

— a behavioural model that captures the overall behaviour of the system functions;

— a temporal model that captures the timing related aspects of the architecture;

— a structural model that captures the system elements and their physical interfaces;

— a mass model that captures the mass related aspects of the system;

— a layout model that captures the absolute and relational spatial placements of the system elements;

— a network model that captures the flow of resources among the applicable system functions or elements.

A system model - is used to represent a system and its environment - may comprise multiple views of the system to support planning, requirements, architecture, design, analysis, verification, and validation - is a representation of a system with various degrees of formalism often expressed as a combination of descriptive and analytical models.

The multiple crosscutting relationships between the model elements contained in a model repository enable the system model to be viewed from many different perspectives to focus on different aspects of the system. The composition of multiple views supports planning, requirements, design, analysis, and verification.

A primary use of the system model is to enable the design of a system that satisfies its requirements and supports the allocation of the requirements to the system’s components (hardware or software). A system model identifies the components, at a particular abstraction level, as black boxes and addresses their interfaces, behaviour, performances and qualities, but not model their internal design. However, the system models are equally valuable throughout the life cycle: during deployment, operation, maintenance, etc.

The system model is an integrating framework for other models and development artefacts including text specifications, engineering analytical models, hardware and software design models, and verification models. In particular, the system model relates the text requirements to the design, provides the design information needed to support analysis, serves as a specification for the hardware and software design models, and provides the test cases and related information needed to support verification and validation.

This is deemed as being the core model. Such system models shall:

— serve at least one engineering process (cf. perform MBSSE or Perform “an engineering process”);

— fulfil a clear and defined purpose involving identified stakeholders;

— follow the applicable rules, principles and guidance of the model owner organization.

Systems models can be dedicated to (and not restricted to) the following features (or characteristics) of systems or objects under consideration:

— its structure;

— its behaviour;

— their characteristics, qualities, attributes;

— their boundaries.

NOTE Concerns or features of interest can be maintained in the model repository by a deployment.

Systems models can also match various perspectives such as:

— system “as dreamed”;

— system “as perceived”;

— system “as contracted”;

— system “as designed”;

— system “as built”.

A system model itself can be described by:

— its context which aids its understanding;

— its purpose which can be distinct from the life cycle of the entity of interest to be modelled;

— its life cycle which can be distinct from the life cycle of the entity of interest to be modelled.

7.2.1.2 Inputs

The following inputs should be available to perform the ‘produce system models’ process:

a) MBSSE planning for the “build models” including:

   1) purposes and Goals for the model and associated measures;

   2) identified features of interest and stakeholders concerns of the systems to consider;

   3) selected formalisms or languages for model representation.

b) required engineering data coming from the business and domain specialists;

c) modelling guidelines and rules;

d) knowledge reuse about or from elements: models, methods and tools.

7.2.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the ‘produce system models’ process.

a) Engineering data are collected

b) Systems models are produced.

c) Model views and perspectives are selected.

d) Capability to produce model views is available.

e) Model capabilities to generate artefacts usable by non-modellers (tables, charts, presentations, text) on demand are defined.

f) Analysis results are considered.

g) Data to support decision-making by relevant stakeholders are selected.

7.2.1.4 Tasks

The organization shall implement the following tasks with respect to the ‘produce system models’ process.

— Collect engineering data.

— Build descriptive models.

— Build analytical models.

7.2.2 Collect engineering data

The goal of this task is to collect required engineering data from the project stakeholders or domain specialists. While the tasks that follow are common approaches as the transition to MBSSE is taking place, it is anticipated that eventually, this task of collecting engineering data will be phased out. In the future, the process of building models and performing MBSSE will simultaneously result in the collection of the engineering data described in this clause.

a) The method should support the goal to collect engineering data with the following capabilities:

   1) collecting required data from the stakeholders.

b) Tool capabilities should support the goal to collect engineering data by allowing the user to do the following:

   1) provide a mechanism based on ontology and using concepts familiar to the stakeholders and defined within the domain ontology;

   2) implement the mapping between domain ontology and system meta-models and also to the meta-model of the tool;

   3) provide data import tools in order to import the data from templates into the model repository.

7.2.3 Build descriptive models

The goal of this task is to build the system models incrementally using collected engineering data.

a) The method should support the production of the system model with the following capabilities:

   1) identifying and analysing the concepts, their relationships, and the structure of the system-of-interest;

2) identifying and formalizing the operational needs and missions;

3) identifying and formalizing the system environment, capabilities, boundary or interfaces;

4) identifying and formalizing the system logical structure or architecture;

5) refining the system capabilities according to its logical structure or architecture;

6) identifying and formalizing the candidate system organic or physical structures or architectures;

7) identifying and formalizing the breakdown structure (i.e. identify configuration items (CIs)) with reference to the integration strategy (i.e. integration level of responsibility, CIs time of deliveries, etc.).

b) Tool capabilities should support production of the system model goals by allowing the user to do the following:

1) model language support to produce different views;

2) model rules and consistency check implementation;

3) maintain the consistency and coherency of the system models;

4) automate report generation;

5) track progress;

6) keep a history of the decisions made;

7) help ensure the accuracy and configuration of the information;

8) verify the model meets rules;

9) execute models to help ensure they have no logic errors.

7.2.4 Build analytical models

The goal of this task is to build the analytical models required to analyse the system.

a) The method should support the goal to build analytical models with the following capabilities:

1) producing the analytical models;

2) analysing the system structure or architecture static and dynamic or behavioural properties (formalized in the descriptive models) with regards to operational needs and missions, specified performances and MBSSE modelling goals;

3) identifying the best architectural compromise or trade-off taking into account all system aspects, including safety, mass, cost, etc.

b) Tool capabilities should support the goal to build analytical models by allowing the user to do the following:

1) language support to produce the analytical models;

2) model simulation support including tools and data sets;

3) model execution support;

4) create the simulation models for:

i) the virtual simulations representing a system existing and operating in its environments throughout the relevant part of its life cycle;

ii) the virtual simulations representing system design both physically and logically;

iii) the constructive simulations representing a system and its employment;

iv) the live simulations, which are simulated operations with real operators and real equipment.

5) support to realize mock-up or user interfaces;

6) automate report generation;

7) track progress;

8) keep a history of the decisions made.

7.3 Produce discipline-specific models

7.3.1 Principal constituents

7.3.1.1 Purpose

The purpose of this process is to elaborate discipline-specific models such as safety, mechanical, thermal, and software models which each address specific engineering constraints and concerns.

NOTE A combination of discipline-specific models and the integration between them needs to be considered.

In a systems engineering approach, discipline-specific models can be dedicated to (and not restricted to) the following aspects of systems or objects under consideration:

— mechanical aspects;

— thermal aspects;

— electrical aspects;

— radio frequency aspects;

— electronic aspects;

— software aspects;

— economical or business aspects;

— financial aspects;

— legal aspects;

— political aspects;

— sociological aspects;

— cultural, historical, geographical aspects;

— organizational aspects;

— dependability aspects;

— resilience aspects;

— safety aspects;

— security aspects;

— sustainability aspects;

— all other relevant quality attributes (“ilities”).

As for system models, discipline-specific models can also match various perspectives.

7.3.1.2 Inputs

The following inputs should be available to perform the ‘produce discipline-specific models’ process:

a) identified aspects to consider;

b) acquisition of interfaces to discipline-specific tools integration;

c) selected formalism or language for model representation;

d) modelling guidelines and rules;

e) knowledge reuse about or from elements: models, methods and tools;

f) system models.

7.3.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the ‘produce discipline-specific models’ process.

a) Engineering data are collected.

b) Discipline-specific models are produced.

c) Interfaces between the system models and existing discipline-specific tools and models are developed.

7.3.1.4 Tasks

The organization shall implement the following tasks with respect to the ‘produce discipline-specific models’ process.

— Collect engineering data.

— Build discipline-specific models.

— Develop the interfaces between the System models and existing discipline-specific tools and models.

7.3.2 Collect engineering data

The goal of this task is to collect required engineering data from the project stakeholders or domain specialists.

a) The method should support the goal to collect engineering data with the following capabilities:

    1) collecting required data from the discipline specialists.

b) Tool capabilities should support the goal to collect engineering data by allowing the user to do the following:

    1) provide data import tools in order to import the data from templates into the specific modelling tools or repository.

7.3.3 Build discipline-specific models

The goal of this task is to build discipline-specific models.

a) the method should support the same capabilities as for building system models. However here the following approaches are possible:

   1) enriching or augmenting system models;

   2) building separate discipline-specific models;

   3) any combination of these approaches can be used.

   4) integrating relevant discipline-specific models;

   5) combining relevant discipline-specific models.

b) Tool capabilities should support the goal to build aspect or discipline-specific models by allowing the user to do the following:

   1) enrich and augment models;

   2) build discipline-specific models;

   3) rely on and archive legacy data;

   4) combine discipline-specific models;

   5) integrate discipline-specific models.

7.3.4 Develop the interfaces between the system models and existing discipline-specific tools and models

The goal of this task is to develop the interfaces between the system models and existing discipline-specific tools and models.

a) The method should support the development of interfaces between the system models and existing discipline-specific tools and models. However here three main approaches are possible:

   1) enriching or augmenting system models;

   2) alternatively, building separate models;

   3) establish traceability or other relations between model interfaces.

b) Tool capabilities should support the development of the interfaces between the system models and existing discipline-specific tools and model goals by allowing the user to do the following:

   1) enrich, augmenting the models;

   2) build separate models;

   3) keep historical data.

NOTE 1 In some cases, the development of discipline-specific models requires contributions from other disciplines.

NOTE 2 In some cases, the development of interfaces between discipline-specific tools and models are also required.

7.4 Verify models

7.4.1 Principal constituents

7.4.1.1 Purpose

The purpose of this process is to define how to help ensure that the models satisfy the model objectives and requirements that represent the stakeholders' needs and comply with conventions described by modelling guides. This process uses automated checking, reviews by relevant stakeholders or both.

NOTE In some cases, conventions include rules, syntactic or grammatical; other conventions make take other forms.

7.4.1.2 Inputs

The following inputs should be available to perform the 'verify models' process:

a) verifiable MBSSE modelling objectives and requirements that capture stakeholders needs;

b) MBSSE modelling verification criteria;

c) models under construction;

d) MBSSE modelling guide to provide the conventions to be verified;

e) modelling rules, syntax, semantics and quality metrics.

7.4.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the 'verify models' process.

a) Verification results are established.

7.4.1.4 Tasks

The organization shall implement the task 'verify models' with respect to the verify models process.

7.4.2 Verify models

The goal of this task is to verify models.

a) The method should support, for some model types, but not generally, the verification of models with the following capabilities:

    1) maintaining models internal and external consistency with reference to the defined strategy and guidance or rules and the collaborative engineering decisions;

    2) maintaining models integrity;

    3) verifying models against model requirements;

    4) making corrective actions based on the verification results;

    5) evaluating MBSSE activities and products against quality and progress metrics and success criteria;

    6) evaluating model conformance against identified methodological rules, formalizing rules, patterns, etc.;

   7) assessing model level of detail and accuracy against modelling objectives or targeted usages.

b) Tool capabilities should support the verification of models by allowing the user to do the following:

   1) define the verification rules;

   2) evaluate the models for compliance to the verification rules and model requirements.

NOTE The typical approach to checking model use consistency is based on the following levels.

— Well-formedness: Does the model conform to its conventions or its Meta model?

— Verification: Is the model interpretable by its stakeholders?

7.5 Validate models

7.5.1 Principal constituents

7.5.1.1 Purpose

The purpose of this process is to help ensure that the models fulfil the needs of the relevant stakeholders.

NOTE One way to validate the models is to help ensure that the modelling goals and objectives established by stakeholders during the planning phases are indeed met by the models that have been built; the necessary decisions have been supported by the model; the required artefacts/outcomes have been produced.

7.5.1.2 Inputs

The following inputs should be available to perform the ‘validate models’ process:

a) documented user needs (higher level requirements document, concept of operation (ConOps) or statement of objectives);

b) stakeholders' acceptance criteria to fulfil their needs;

c) models to be validated;

d) MBSSE management plan stating the expected levels of confidence of the models.

7.5.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the ‘validate models’ process.

a) Level of confidence of the models is analysed.

b) Model behaviour is assessed with regard to expectations.

c) Validation results are established.

c) Model improvements are identified to meet expectations.

7.5.1.4 Tasks

The organization shall implement the following tasks with respect to the ‘validate models’ process.

— Validate models.

7.5.2 Validate models

The goal of this task is to validate the models.

a) The method should support the validation of the models with the following capabilities:

   1) ensuring that the system, represented by its model(s), meet expectations;

   2) enabling the isolation of a model of a system from the other models - to help ensure the level of "quality" of what a model represents independent of other models of the same system - to improve confidence in a model of the system;

   3) enabling the composition and/or federation of models of the system taking into account their differences (semantics, level of maturity, modelling language used, level of detail addressed by the system);

   4) enabling the building of a coherent and, if possible, complete multi-view representation with the objectives of the actors;

   5) enabling the use of the models to produce simulations, non-functional property evaluations, various analyses such as sensitivity or dependence analysis, impact analysis, effects propagation analysis, etc.; and interpret them for the purpose of decision support evidence;

   6) ensuring that the system as modelled (i.e. represented by a set of heterogeneous models, federated or simply associated), interacts as desired with the systems present (continuously or may appear according to certain situations) in its different operational contexts (i.e. concept of operations) and in all operational situations;

   7) ensuring that the system as modelled, as changes are introduced but not only, remain a sufficient representation of the system for the intended purpose, stakeholder expectations and needs such as higher-level requirements, operational concepts (OpsCon), etc.

b) Tool capabilities should support the validation of the models by allowing the user to do the following:

   1) allow the analysis of models for system validation and confidence;

   2) represent the set of heterogeneous, federated or simply associated models;

   3) check that the models meet all the requirements that are allocated to it in all the operational situations that have been specified;

   4) check the integrity of the system regarding non-functional properties.

7.6 Simulate systems using models

7.6.1 Principal constituents

7.6.1.1 Purpose

The purpose of this process is to support the different steps of MBSSE through the production of information and data representative of the system under engineering – as modelled and parameterized at the considered step – in order to analyse it under different points of view to support into deliberation on decisions about alternatives or parameter refinements.

This process participates in the validation of the system. This allows two possibilities, on the one hand to validate that the model(s) is (are) well / sufficiently representative (and therefore of confidence / confidence) of the system for the level considered and on the other hand that the model(s) is (are) sufficiently reliable to participate in the validation of the system

This process is useful for testing, analysis or training where real-world systems or concepts are represented by a model then by simulating those models. A simulation implements a model over time, so it brings a model to life and shows how a particular object or phenomenon behaves. The main purpose of a simulation is to gain system understanding without manipulating the real system, either because it is not yet defined or available or because it cannot be executed directly due to cost, time, resources, or risk constraints.

In order to do so, methods and tools should be provided (i) for animating – or simulating – or executing – the system models (basically from a behavioural aspect), by providing a dynamic capability to do so, and (ii) for performing analyses, based on simulation results (and possibly results of other work).

NOTE Generally, due to its nature, model simulation is an operational way to support model verification or model validation (see above).

7.6.1.2 Inputs

The following inputs should be available to perform the 'simulate systems using models' process:

a) systems simulations objectives;

b) overall simulation (and / or elementary components) validity domain;

c) analytical (system) model(s);

d) MBSSE management plan;

e) simulation environment and procedures;

f) simulation strategies;

g) objective and the expected values of simulations;

h) data representative sets;

i) overall simulation (and / or elementary components) validity domain.

7.6.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the 'simulate systems using models' process.

a) Simulation environment is available.

b) Simulation procedure is specified

c) System simulation analyses results are validated.

7.6.1.4 Tasks

The organization shall implement the following tasks with respect to the 'simulate systems using models' process.

— Prepare simulation environment with required data and models.

— Simulate systems using models.

— Analyse results and Validate behaviours.

7.6.2 Prepare simulation environment with required data and models

The goal of this task is to prepare the environment to use the simulation models.

a) The method should support the goal to prepare simulation environment with required data and models with the following capabilities:

   1) identifying the simulation "engine": simulation paradigm, dynamic implementation and interactions, etc.;

   2) specifying properties of a model or a set of models required for model simulation (i.e. simulation requirement);

   3) specifying the data or information (i) necessary for model execution and (ii) possibly provided by model simulation (simulation model input or output);

   4) identifying the simulation outputs (data or information) at each step, of their management;

   5) defining of the frame and scope of relevant data or information potentially accessible through simulation of a model (in strong relation with representativeness and validity domain of the considered model) (output data validity);

   6) assessing of the "simulatability" (i.e. simulation ability) of a model – or a set of models (i.e. simulation verification and validation).

b) Tool capabilities should support the goal to prepare simulation environment with required data and models by allowing the user to do the following:

   1) support for data stores for the input and output data;

   2) provide data import tools;

   3) providing mock-ups, user interfaces to command and control demonstrations or simulations.

7.6.3 Simulate systems using models

The goal of this task is to use different means to demonstrate, exercise simulation using a tool, test the model and ask questions to obtain results and answers.

a) The method should support the simulation of systems using the model with the following capabilities:

   1) specifying the simulation operation, including specifications of:

      i) the objectives of the simulation regarding the systems engineering phase under study (definition of needs or requirements, system requirements, system design, operation, maintenance or disposal);

      ii) the experimentation plan, management, steps and content;

      iii) the simulation outputs (data or information) at each step, of their management;

      iv) the simulation architecture: simulated world boundaries and characteristics, models organization (system under test model(s) vs. system environment models), simulation paradigm(s), architecture dynamics, conceptual or logical or physical interactions and dynamic, etc.;

      v) the simulation environment models: scenarios and related parameters and data, functional coverage and dynamic of these models for providing relevant system models interactions;

      vi) the simulation "engine": simulation paradigm, dynamic implementation and interactions, etc.;

vii) the data or information exploitation: management and processing – concerning the deliberation context, objectives and content: solution verification, explanation, alternatives exploration, etc.

b) Tool capabilities should support the simulation of systems using the model by allowing the user to do the following:

1) build the modelling and simulation development environment – providing relevant modelling paradigm(s) tools – for:

i) development and test of system model(s);

ii) development of multi-scale multi-purpose system environment models;

iii) integration of under test system models and environment ones;

iv) repository for the knowledge reuse of models (including multi-level models) with capabilities of inter-level modelling verification and validation.

2) build the simulation execution environment providing:

i) simulation "engines" for models dynamic simulation and interaction;

ii) probes and measurement tools for required output data capture and management;

iii) supervision tools for simulation overall management, including scenarios building and knowledge reuse tools.

7.6.4 Analyse results and validate behaviours

The goal of this task is to analyse results and validate behaviours.

a) The method should support the goal to analyse results and validate behaviours with the following capabilities:

1) analysing the results and validating the behaviours either for:

i) the virtual simulations representing real people operating simulated systems;

ii) the constructive simulations (simulation of people operating simulated systems) representing a system and its deployment;

iii) the live simulations which are simulated operations with real operators and real equipment.

2) producing analysis reports;

3) realize required model updates following the system analysis.

b) Tool capabilities should support the goal to analyse results or validate behaviours by allowing the user to do the following:

1) analyse the results and validate behaviours with regard to the objectives and the expected value of the simulation;

2) update models and manage changes or versions.

7.7 Make decisions using models

7.7.1 Principal constituents

7.7.1.1 Purpose

The purpose of this process is to enrich model outcomes, take decisions about alternatives and perform parameter refinements using the results of simulation.

This process is supported by trade-off analyses to examine viable alternatives and to determine which is preferred. It is important that there be criteria established and an agreed-upon approach to measuring alternatives against the criteria. In a model-based approach, system model elements are characterized by these criteria when they are considered in alternatives.

See ISO/IEC/IEEE 42020 and ISO/IEC/IEEE 42030.

7.7.1.2 Inputs

The following inputs should be available to perform the 'make decisions using models' process:

a) modelling and simulation strategies;

b) model variants and model elements;

d) criteria for analysis and decisions;

7.7.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the 'make decisions using models' process.

a) Decision on alternatives is made.

c) Rationale for the results is recorded.

7.7.1.4 Tasks

The organization shall implement the following tasks with respect to the 'make decisions using models' process.

— Capture decision criteria within model.

— Generate decision reports (benefit analysis charts, cost-benefit charts, risk-benefit charts).

— Build a rationale.

7.7.2 Capture decision criteria within the model

The goal of this task is to capture decision criteria within the model.

a) The method should support the goal to capture decision criteria within the model with the following capabilities:

   1) capturing decision criteria within the model and examine viable alternatives;

b) Tool capabilities should support the goal to capture decision criteria within the model by allowing the user to do the following:

   1) record the performance of the preferred model with respect to the decision criteria;

7.7.3 Generate decision reports

The goal of this task is to generate decision reports.

a) The method should support the goal to generate decision reports with the following capabilities:

   1) decision analysis based on the decision method and model of choice.

b) Tool capabilities should support the goal to generate decision reports by allowing the user to do the following:

   1) generate the decision reports using the supporting data from the model(s).

7.7.4 Build a rationale

The aim of this task is to build a rationale that supports the selection of the preferred alternative.

a) The method should support the goal to build a rationale with the following capabilities:

   1) specifying properties of a model or a set of models required for the related alternatives;

   2) building a rationale from the results obtained after evaluating the model alternatives.

b) Tool capabilities should support the goal to build a rationale by allowing the user to do the following:

   1) build the evaluation of model alternatives environment:

      i) tools for simulation outputs management, etc.;

      ii) tools for simulation data analysis, including visualization, replay, etc.;

      iii) tools for simulation data processing (statistics, data analysis, etc.), etc.;

      iv) tools for high-level metrics and criteria extraction;

      v) tools for multi-criteria analysis and decision support.

   2) record the rationale.

8 Support models

8.1 General

These processes support core MBSSE processes. They focus on the technical and engineering data management aspects. The associated tools and methods are described. What specific MBSSE approaches should be used to complement traditional engineering activities (such as change management, data management) are identified:

— manage technical quality;

— manage configuration;

— manage data and models;

— share models for collaboration.

8.2 Manage technical quality

8.2.1 Principal constituents

8.2.1.1 Purpose

The purpose of this process is to define how to manage and assess model quality, especially by monitoring and controlling according to plans and how to improve the quality of the system using MBSSE. It helps to ensure that the development processes and artefacts of MBSSE are consistent with the plans, and the predefined quality requirements of an organization.

8.2.1.2 Inputs

The following inputs should be available to perform the 'manage technical quality' process:

a) goals and objectives of technical quality management;

b) quality requirements;

c) defined MBSSE engineering processes;

d) progress and quality metrics;

e) development artefacts of MBSSE (assets, products).

8.2.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the 'manage technical quality' process.

a) A strategy for managing the quality of the MBSSE products is developed.

b) Corrective and preventive actions are managed.

c) Quality assurance results are recorded.

d) Progress and quality metric dashboards are produced.

NOTE The strategy can be documented in a quality management or quality assurance plan.

8.2.1.4 Tasks

The organization shall implement the following tasks with respect to the 'manage technical quality' process.

— Perform technical review.

— Perform quality assurance.

8.2.2 Perform technical review

The goal of this task is to conduct MBSSE reviews at similar points in the design life cycle as with a traditional systems engineering approach.

a) The method should support the goal to perform technical reviews with the following capabilities:

   1) defining technical review process;

   2) identifying design life cycle checkpoints;

   3) performing modelling aspects oriented review;

4) providing guidance for review completion assurance.

b) Tool capabilities should support the goal to perform technical reviews by allowing the user to do the following:

1) share the technical review expectations and result with relevant participants.

8.2.3 Perform quality assurance

The goal of this task is to perform quality assurance.

a) The method should support the quality assurance goal with the following capabilities:

1) establishing the MBSSE quality management plan and quality requirements;

2) recording progress and quality metrics;

3) managing issues on management and exploitation of models;

4) performing issues management and corrective process (identify, communicate and resolve noncompliance issues);

5) establishing records of quality assurance activities for MBSSE.

b) Tool capabilities should support the quality assurance goal by allowing the user to do the following:

1) keep evidence of quality requirements fulfilment;

2) build and display metric dashboards.

8.3 Manage configuration

8.3.1 Principal constituents

8.3.1.1 Purpose

The purpose of this process is to describe how to perform configuration management (CM) of models and use model data to improve CM for the system or software of interest.

Model configuration items (MCIs) are managed to maintain the integrity of the models. CM help ensures that changes to the model baseline and its supporting evidences occur in a managed, controlled and auditable manner. CM for MBSSE treats modelling (core) assets as variants due, for example, to trade-offs and decisions. In addition, configurations associated with modelling assets evolution should be managed in both time and space. Complex relationships among asset configurations, in both time and space (e.g. distributed database, multi-locations team), during collaborative model sharing, make the CM of modelling assets particularly difficult. These difficulties can be mitigated with the complexity management approaches of the systems. Tools and methods for MBSSE CM shall support definition, development, or operating environment for modelling assets and MBSSE process development artefacts. Moreover, definition, development, or operation of modelling assets can be conducted in parallel in a distributed environment, so MBSSE CM shall define and set up appropriate policies and strategies to cope with it.

A MCI satisfies an end-use function and is designated for independent CM by the product developer and by the customer. A MCI can be defined in different granularities, from an individual fine-grained model element, a set of model elements, to the entire model. A snapshot in time of approved changes to all MCIs for a specific project, program or release is called the model baseline.

The granularity and the organization of the MCIs are an important decision for model life cycle management. The organization is often influenced by the organization of the teams creating the model. Good organization of the model can minimize resource conflicts and the need for complicated merging of changes.

Below are principles that should be applied in order to set up effective CM for models.

— Work products and supporting data: segregate all data generated to be either work products or supporting data according to the role of the contained information in the systems engineering process. Models may appear in both, it is important to know in advance. For example, an architecture model is part of a work product, and contributes to defining a CI, but a behavioural model used for trade-off is a supporting data, and may require data management control.

— A model used for trade-off decision/justification should be recorded against a model baseline.

— In order to avoid having only one CI, the different layers of modelling should be planned, and different CI for each design layer should be managed. The system breakdown structure (SBS) used for design, may be different from the manufacturing product breakdown structure(s) (PBS) or bill of material (BOM). The layered approach is a way to better identify the change impact perimeter when a change is internal to a unique system element (and hence limited to associated model elements).

— Planning the engineering baselines: configuration baselines should be defined with respect to their purpose and their expected work product content during project set-up. That includes the models to be authored, and their maturity assessment. As model maturity assessment can be largely automated, baseline maturity can also be largely automated, allowing real-time evaluation of the work progress.

— Change management. One should pay a lot of attention to model interdependency. Change in a model may affect another model. Although this can be tracked, not all tool suites may provide the appropriate dependency management features.

— Change request analysis should make an extensive use of model and simulation, in order, for example, to help ensure that stakeholder requirements fulfilment is not impacted.

NOTE Models produced can be tagged as draft, to be reviewed, and validated or by giving a progress percentage per each model. This allows an estimate of the model maturity by means of the number of views or specific models in a particular state over the number of views and models required. Number of required models are defined for each process within the overall model management plan.

8.3.1.2 Inputs

The following inputs should be available to perform the 'manage configuration' process:

a) CM policies;

b) artefacts used and produced during model engineering.

8.3.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the 'manage configuration' process.

a) CIs related to modelling assets are identified.

b) Configuration baselines are established.

c) MBSSE configuration breakdown structure is established.

d) MBSSE configuration breakdown structure is controlled.

e) Configuration status accounting (CSA) is available: CSA is the capture and storage of configuration information (data).

f) Changes are formally controlled.

8.3.1.4 Tasks

The organization shall implement the following tasks with respect to the 'manage configuration' process.

— Manage modelling assets and configuration items.

— Manage changes to models.

8.3.2 Manage modelling assets and configuration items

The goal of this task is to manage modelling assets as the essential piece of data that document configuration items. Nevertheless, unlike documents that can be managed individually, models are much more complex to control. That is why CM planning requires attention in an MBSSE environment.

a) The method should support the goal to manage modelling assets and CIs with the following capabilities:

   1) identifying circular, many-to-many, and suspect traces between MCIs;

   2) defining and maintaining the configuration breakdown structure including relationships among structure elements;

   3) establishing trace and control mechanism for each breakdown structure, for configuration of each of the layers, and for the relationship between the two of them;

   4) creating or releasing baselines of modelling assets;

   5) identifying configurations items related to modelling assets;

   6) informing, managing and tracing the differences among the configurations items;

   7) performing CSA: the CSA contains a record of the status (CI Record or CIR) of pending, approved and embodied changes to the modelling assets or CI and can therefore provide a picture of the CI at any given time.

b) Tool capabilities should support the goal to manage modelling assets and CIs by allowing the user to do the following:

   1) allow the definition of modelling assets configurations;

   2) allow the definition of a breakdown structure;

   3) support traces among breakdown structure elements in accordance with the differences among modelling assets;

   4) provide capability for tracing each modelling assets configuration from the breakdown structure;

   5) support capability for tracing each modelling asset's configuration in time dimension.

8.3.3 Manage changes to models

The goal of this task is to perform the change management of models.

a) The method should support the change management of models with the following capabilities:

   1) identifying and recording change requests for Model(s) configuration;

   2) evaluating and coordinating the impacts of change requests (e.g. compare models, analyses change impacts on models);

   3) defining baselines for CM;

    4) assessing deltas between several versions of models;
    5) defining the statuses applicable for CIs.

b) Tool capabilities should support the change management of model goals by allowing the user to do the following:
    1) manage the changes and issues;
    2) provide impact analysis capability;
    3) make model baselines description according to the change requests set.

8.4 Manage data and models

8.4.1 Principal constituents

8.4.1.1 Purpose

The purpose of this process is to consider the efficient data management and describe the data-manipulation in models. It includes data and model life cycles, exchange and vertical continuity through system levels. It also defines activities, methods, and tool-capabilities for ensuring engineering data consistency in the model repository.

The data concerned by this process are model elements and associated meta-data. These model elements can reference each other and the users can navigate between model elements using the references.

Model life cycle management synchronize the CRUD (create, read, update, and delete) operations on heterogeneous models within the supporting modelling tools and model repositories, throughout the system development life cycle. This is accomplished through the management of versions, variations, configurations and baselines of models, simulations, analysis results, and the tools that are used by multiple geographically dispersed users. In addition, all the metadata associated with the models, tools, and analysis results including who made the change, what changes were made, when and why, as well as information regarding the application of the model are included in model life cycle management.

Model management systems should be able to leverage current data management practices. Data management is focused on ensuring the integrity of data generated for each operational group of a business and is a major part of the basic capabilities of database management systems (DBMS). Data management teams work concurrently with other management teams at the enterprise and project levels to define appropriate control processes for identified data sets.

8.4.1.2 Inputs

The following inputs should be available to perform the ‘manage data and models’ process:

a) data management policy;
b) MBSSE management plan;
c) MBSSE CM plan including MCIs.

8.4.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the ‘manage data and models’ process.

a) Data and Models management policy is established.
b) Infrastructure to support data and model management is defined.

8.4.1.4 Tasks

The organization shall implement the following tasks with respect to the 'manage data and models' process.

— Define the modelling management policy.

— Define infrastructure needs to support data and model management.

8.4.2 Define the data and models management policy

The goal of this task is to define the data and models management policy.

a) The method should support the definition of the data and models management policy with the following capabilities:

   1) defining the main rules of the policy to modelling elements;

   2) defining the rights, obligations, and commitments of parties for generation, management, access privileges, and sharing criteria;

   3) identifying the standards by which the data and information are created, managed and stored. These standards enable the integration and sharing of the data and information contained in the integrated, shareable sets of data.

b) Tool capabilities should support the definition of the data and models management policy by allowing the user to do the following:

   1) implement the main rules;

   2) configure sources of data and information and designating authorities (owners) and responsibilities regarding the origination, generation, capture, archival, sharing, and disposal of information in accordance with the records and CM process and governing standards and requirements.

8.4.3 Define infrastructure needs to support data and model management

The goal of this task is to define the infrastructure needs to support data and model management.

a) The method should support this task with the following capabilities:

   1) identifying model and data management tools and processes, as well as methodologies, standards, and procedures;

   2) developing, establishing, and managing enterprise data architecture and platforms in alignment with model and data governance and management policies, principles, processes, and requirements defined;

   3) supplying, maintaining, and providing support for hardware and software (SE toolsets) needed to meet the needs of the model and data management activities;

   4) designing and implementing model and data access, security, search, sharing, quality, backup, and archival storage control services in alignment with the model and data governance;

   5) training systems engineers in the use of the data management toolset and processes.

b) Tool capabilities should support this task by allowing the user to do the following:

   1) define a master schema for the integrated, shareable sets of data and databases to store the data associated including formats and media for capture, retention, transmission, and retrieval of model and data.

8.5 Share models for collaboration

8.5.1 Principal constituents

8.5.1.1 Purpose

The purpose of this process is to define how to work on a system or a software model in a distributed environment concurrently and how to share models.

The challenges of the collaborative modelling are multi-dimensional, as the system model is enriched by different types of models being developed by different users that are often geographically distributed. The models are being updated by different users at different times. The modellers may use different tools and the tool revision may also change over time.

There are important aspects to consider including security, data integrity, copyright, intellectual property rights (IPR) etc. while performing collaborative modelling.

Models may be developed in independent databases or a single consolidated database operated as a collaborative workspace for remotely located teams of authors in independent enterprises. Models and supporting data should be identified with the level of control required according to the configuration and data management plans governing the project. Data managers continually oversee data security, assign access controls as defined by the identified project life cycle controls, and help ensure data integrity throughout the projected retention period of the completed data sets.

8.5.1.2 Inputs

The following inputs should be available to perform the 'share model for collaboration' process:

a) MBSSE management plan including team organization;

b) constraints to be respected while collaborative modelling and model sharing;

c) CM plan;

8.5.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the 'share models for collaboration' process.

a) Collaborative modelling guidelines are shared.

b) Collaborative modelling environment (tool chains, tool user guides) is defined.

c) Model sharing and authoring rules are defined.

8.5.1.4 Tasks

The organization shall implement the following tasks with respect to the 'share model for collaboration' process.

— Define collaborative modelling guidelines and environment.

— Define model sharing and authoring rules.

— Maintain the consistency of models.

8.5.2 Define collaborative modelling guidelines and environment

The goal of this task is to provide the guidelines for collaborative modelling and define the required environment including tool chains and the user guides.

a) The method should support this task with the following capabilities:

1) identifying collaborative team organization;

2) specifying collaborative modelling environment (tools, connections between tools);

3) developing collaborative environment user guides.

b) Tool capabilities should support this task by allowing the user to do the following:

1) set up modelling environment and test;

2) provide easy access to the user guides;

3) have controlled access to the collaborative modelling environment.

8.5.3 Define model sharing and authoring rules

The goal of this task is to define rules for the model sharing and authoring including: security, data integrity, copyright, IPR etc.

a) The method should support this task with the following capabilities:

1) defining model sharing and authoring rules.

b) Tool capabilities should support this task by allowing the user to do the following:

1) have controlled access to the models.

8.5.4 Maintain the consistency of models

The goal of this task is to help ensure the consistency of models within the collaborative environment and shared models.

a) The method should support this task with the following capabilities:

1) performing regular consistency checking;

2) controlling CRUD operations according to the guidelines.

b) Tool capabilities should support this task by allowing the user to do the following:

1) automate consistency checking;

2) maintain the integrity or consistency to the models.

9 Perform MBSSE

9.1 General

Perform MBSSE describes how to carry out the model-based systems and software engineering activities using models. The perform MBSSE processes depend upon the objectives of MBSSE and the context. The "perform MBSSE processes are scoped, managed, monitored and supported by the processes defined within other three main process groups. Thanks to the models, the system or software of interest may be analyzed throughout its life cycle; the model(s) can be executed (for maintaining integrity, or consistency) to discover errors early, gain understanding and to validate concepts.

The technical management and technical processes of ISO/IEC/IEEE 15288 apply to this document.

Not all processes are used while modelling a given system or software. The processes to be performed are identified during MBSSE scoping and planning. The Perform MBSSE group includes the following processes:

— perform business and mission analysis;

— perform operational analysis;

— perform functional analysis;

— perform system structure design;

— perform system analysis;

— perform domain design integration;

— perform system verification and validation;

NOTE 1 The tasks describing all the processes above can be supported by some common tool capabilities:

— modelling language support preferably based on an appropriate meta model;

— providing templates and rules to facilitate the selected method, framework;

— automating model verification and support validation;

— CM support for the model artefacts;

— document generation from model.

— web based model reviews, inspections.

NOTE 2 Only the additional tool capabilities are described within each task.

9.2 Perform business and mission analysis

9.2.1 Principal constituents

9.2.1.1 Purpose

The purpose of this process is to establish the enterprise architecture model. The enterprise can be driven by one or more organizations and the stakeholders can belong to several organizations. The analysis of the business helps to identify the opportunities. The analysis of the enterprise mission allows the modelling of enterprise goals, definition of mission success criteria, modelling of mission environment, the identification of the required capabilities, organizational structures and the solution overview.

9.2.1.2 Inputs

The following inputs should be available to perform this process:

a) enterprise vision and mission statements;

b) problems to solve;

c) stakeholders, their needs and expectations;

d) business and mission plans;

e) modelling strategies;

f) decision analysis criteria.

9.2.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the 'perform business and mission analysis' process.

a) The model composed of a high-level a target architecture which includes preliminary ConOps, mission scenarios, services and technology maps and gap analysis compared to the existing architecture is defined.

b) Business and mission requirements and business and mission success criteria are agreed by stakeholders.

c) High-level ConOps, business and mission requirements, etc. are established and deliverable documents are generated from the models according to the tool setup required by the specific project or program.

d) Capabilities and roadmaps are established.

e) Model progress and review dashboards are defined.

f) Model maturity is evaluated.

9.2.1.4 Tasks

The following tasks are performed with respect to the 'perform business and mission analysis' process.

— Describe high-level target enterprise architectures using models.

— Evaluate candidate architectures and analyse gaps using models.

— Establish capability roadmaps.

— Define business and mission requirements.

— Generate ConOps.

9.2.2 Describe high-level target enterprise architectures using models

The goal of this task is to analyse problems, discover opportunities and establish target enterprise architectures. To solve a problem several candidate architectures can be conceived.

a) The method should support the goal to describe high-level target enterprise architectures using models with the following capabilities:

   1) creating goals, associated benefits, and required capabilities;

   2) linking goals and capabilities;

   3) defining the capability phasing;

   4) establishing candidate architectures consisting of different technologies and services (i.e. variant modelling);

   5) modelling mission contexts, phases, scenarios and associated mission success criteria;

   6) modelling mission environment.

b) Tool capabilities should support the goal to describe high-level target enterprise architectures using models by allowing the user to do the following, in addition to the common tool capabilities:

   1) provide dashboards.

9.2.3 Evaluate candidate architectures and analyse gaps using models

The gaps between existing and target architectures should be analysed in order to establish roadmaps.

The goal of this task is to assess the candidate architectures to help in selecting the most suitable one.

a) The method should support the goal to analyse gaps and evaluating candidate architectures using models with the following capabilities:

   1) performing trade-off between candidate architectures;

   2) analysing gaps between existing and target architectures.

b) Tool capabilities should support the goal to analyse gaps and evaluate candidate architectures using models by allowing the user to do the following:

   1) gap analysis templates and tables;

   2) have mechanisms to manage several candidates architectures;

   3) have a trade-off method;

   4) have multi-criteria analysis and decision support.

9.2.4 Establish capability roadmaps

The capability roadmaps are defined to smoothly move the enterprise architecture towards the target.

a) The method should support the establishment of capability roadmaps with the following capabilities:

   1) organizing captured capabilities into roadmaps with provisioning technologies and services, by taking into account schedule, budget, resource constraints;

   2) sharing and updating roadmaps.

b) Tool capabilities should support the establishment of capability roadmaps by allowing the user to do the following:

   1) present capabilities as Gantt charts and update them;

   2) share these roadmaps in a tool agnostic way (i.e. web based user interface).

9.2.5 Define business and mission requirements

The business and mission requirements can be specified and derived from the high-level architectures established. Mission scenario modelling, especially, helps to clarify mission requirements.

a) The method should support the definition of business and mission requirements with the following capabilities:

   1) creating the requirements and their traceability to the corresponding model elements.

b) Tool capabilities should support the definition of business and mission requirements by allowing the user to do the following:

   1) manage the business and mission requirements and their traceability.

9.2.6 Generate ConOps

The ConOps may contain the graphical and textual descriptions.

a) The method should support the generation of ConOps with the following capabilities:

   1) defining graphical and textual model contents using a meta-model;

   2) managing the document structure and linked model elements within a repository.

b) Tool capabilities should support the generation of ConOps by allowing the user to do the following:

   1) implement rules to verify the conformance of the contents to the meta-model;

   2) automate ConOps document generation.

9.3 Perform operational analysis

9.3.1 Principal constituents

9.3.1.1 Purpose

The purpose of this process is to define the OpsCon which is prepared initially to support the concept and development stages of the system life cycle and then maintained throughout the Program to support the production, utilization, support, and retirement stages. Stakeholders and theirs needs differ according to the system context which itself depend on the system life cycle stage. Models help to share a common understanding between stakeholders and to formalize the needs into Stakeholder requirements.

9.3.1.2 Inputs

The following inputs should be available for the 'perform operational analysis' process:

a) ConOps;

b) services and technology maps and gap analysis compared to existing architecture;

c) mission scenarios, Business and mission requirements and Business and mission success criteria;

d) capabilities and roadmaps;

e) model progress and review dashboards;

f) model maturity evaluation.

9.3.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the 'perform stakeholder needs analysis and early validation' process.

a) System life cycle, including stages is defined.

b) System context and environment description and milestones are defined.

c) OpsCon including use case scenarios is described.

d) Other preliminary life cycle concepts are established.

e) Stakeholder requirement description, validation criteria, measures of effectiveness (MOEs) are defined.

f) Validation scenarios are defined.

g) User interface mock-ups and simulation models are built.

9.3.1.4 Tasks

The organization shall implement the following tasks to 'perform operational analysis'.

— Identify system life cycle, boundary and context.

— Identify stakeholders.

— Identify use cases and develop use case scenarios, validation scenarios.

— Identify operational modes.

— Capture stakeholder requirement and MOEs.

9.3.2 Identify system life cycle, boundary and context

The goal of this task is to define the system context and system life cycle in order to establish the models for OpsCon and other preliminary life cycle concepts.

a) The method should support the identifying of system life cycle, boundary and context with the following capabilities, in addition to the common tool capabilities:

   1) modelling system life cycle;

   2) modelling system context by defining the boundary, elements within environment external to the system-of-interest and the interaction with those elements;

   3) realizing preliminary life cycle concept (OpsCon, utilization concept, disposal concept, etc.) models.

b) This task should be supported by the common tool capabilities.

9.3.3 Identify stakeholders

The goal of this task is to identify all the stakeholders having an interest in the system, classify them according to their interest, and influence in order to define and prioritize the requirements.

a) The method should support the identifying of stakeholders with the following capabilities:

   1) discovering and involving stakeholders;

   2) interviewing stakeholders;

   3) capturing stakeholder interests;

   4) classifying stakeholders.

b) Tool capabilities should support the identifying of stakeholders by allowing the user to do the following, in addition to the common tool capabilities:

   1) provide support and templates to capture and classify stakeholders and their interests.

9.3.4 Identify use cases and develop use case scenarios, validation scenarios

The goal of this task is to build models to share common understanding of the way of using the system-of-interest.

The use case analysis is a term originated from software domain and widely used by system engineers to designate operational analysis. The operational analysis is not intended to describe the solution or the functions but the usage of functions offered by the system of interest.

a) The method should support the modelling of use cases and scenarios including validation scenarios with the following capabilities:

   1) producing use case diagrams for each life cycle stage according to the modelling objectives (utilization, maintenance, disposal, etc.);

   2) analysing, refining and completing operations identified and modelled within mission scenarios;

   3) modelling use case scenarios, in the way that they can be used for early validation (execution or simulation) and to support validation plans.

b) Tool capabilities should support the modelling of use cases and scenarios including validation scenarios by allowing the user to do the following, in addition to the common tool capabilities:

   1) support to generate validation plans.

9.3.5 Identify operational modes

The goal of this task is to define the operational modes, the way of transition between modes and how the system is operated within different modes.

a) The method should support the identification of operational modes with the following capabilities:

   1) modelling operational modes and transitions (i.e. state machines);

   2) linking the scenarios to operational modes.

b) This task should be supported by the common tool capabilities.

9.3.6 Capture stakeholder requirements and measures of effectiveness (MOEs)

The goal of this task is to formalize the stakeholder needs and constraints into stakeholder requirements and define MOEs to use within validation.

a) The method should support the capturing of stakeholder requirements and measures of effectiveness (MOEs) with the following capabilities:

   1) analysing stakeholder needs and constraints using operational models;

   2) capturing formalized stakeholder requirements;

   3) defining and capturing MOEs.

b) Tool capabilities should support the capturing of stakeholder requirements and measures of effectiveness (MOEs) by allowing the user to do the following, in addition to the common tool capabilities:

   1) support to capture requirements and manage traceability.

9.4 Perform function analysis

9.4.1 Principal constituents

9.4.1.1 Purpose

The purpose of this process is to define the solution or system in terms of its functions, dysfunctions and performances. The model provides the basis for defining requirements and verification.

In order to meet its requirements, a system has to deal with a set of responsibilities related to purpose. Typically, each system function has a number of uses associated with it and an understanding for each of these define a system functional requirement with associated performance criteria.

The functional flow models are used to generate system requirements, and verification methods. This is one key area where the majority of modelling is performed. Several methods, frameworks, and tools exist in this group.

9.4.1.2 Inputs

The following inputs should be available for the 'perform function analysis' process:

a) system context, environment description and the life cycle including stages and milestones;

b) OpsCon including use case scenarios which includes expected system services;

c) other preliminary life cycle concepts;

d) stakeholder requirement description including validation criteria, measures of effectiveness (MOE).

9.4.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the 'perform function analysis' process.

a) Functional break down and functional flows are defined.

b) Potential dysfunctions are identified.

c) Function to operational activity traceability is established.

d) Function allocation to high-level system elements is established.

e) System requirements, measure of performance (MOP), verification means and requirement traceability are defined.

9.4.1.4 Tasks

The organization shall implement the following tasks with respect to the 'perform function analysis' process.

— Realize functional analysis and decomposition.

— Detect or identify possible dysfunctions.

— Identify system elements and allocate functions and develop functional flows.

— Capture system requirements, constraints and MOPs.

— Realize and manage traceability.

9.4.2 Realize functional analysis and decomposition

The goal of this task is to identify required functions and establish a hierarchical view. The hierarchical view helps to manage functions.

a) The method should support the realizing of functional analysis and decomposition with the following capabilities:

   1) extracting system services from use case (UC) scenarios and identify required high-level system functions;

   2) decomposing them using appropriate functional analysis techniques such as function analysis system technique (FAST) or functional flow block diagram (FFBD).

b) Tool capabilities should support the realizing of functional analysis and decomposition by allowing the user to do the following, in addition to the common tool capabilities:

   1) have functional analysis techniques.

9.4.3 Detect or identify possible dysfunctions

The goal of this task is to detect or identify possible dysfunctions and model them in order to specify requirements to control them.

This task addresses the important considerations of loss-driven systems engineering speciality areas such as resilience, safety, security, etc.

a) The method should support the detection or identification of possible dysfunctions with the following capabilities:

   1) defining quality objectives as safety and reliability;

   2) identifying system dysfunctions from use case alternate flows and what-if scenarios;

   3) producing dysfunctional and control models.

b) This task should be supported by the common tool capabilities.

9.4.4 Develop functional flows and system states

The goal of this task is to develop functional flows, identify system states and active functional flows within different states.

a) The method should support the developing of functional flows and system states with the following capabilities:

   1) developing functional flows;

   2) developing system state transitions and active functional flows within different states.

b) Tool capabilities should support identify system elements and allocate functions and develop functional flows by allowing the user to do the following:

   1) support in development of functional flows;

   2) support in development of system state transitions.

9.4.5 Capture system requirements, constraints and measure of performance (MOPs)

The goal of this task is to define system requirements with the help of functional and dysfunctional models as well as stakeholder needs and constraints.

a) The method should support the goal to capture system requirements, constraints and MOPs with the following capabilities:

   1) formalizing system functional requirements;

   2) formalizing system interface requirements;

   3) formalizing system performance and quality requirements;

   4) capturing measures of performances (MOPs) and other technical measures;

   5) defining verification means.

b) Tool capabilities should support capture system requirements, constraints and MOPs by allowing the user to do the following, in addition to the common tool capabilities:

   1) support to generate requirements from models or capture requirements.

9.4.6 Realize and manage traceability

The goal of this task is to realize and manage traceability between operational and system functional model elements.

a) The method should support the goal to realize and manage traceability with the following capabilities:

   1) tracing system operational activity to functions;

   2) tracing stakeholder requirements to system requirements.

b) Tool capabilities should support the goal to realize and manage traceability by allowing the user to do the following:

   1) support and document traceability.

9.5 Perform system structure design

9.5.1 Principal constituents

9.5.1.1 Purpose

The purpose of this process is to structure the solution or system by grouping its functions. It includes the design of interfaces, physical structures and qualities. It provides data for specialty engineering analysis. This process is applied recursively at the element level as elements are also considered systems. The model provides the basis for understanding how to describe an understanding of complex problems into a set of smaller and simpler ones while still ensuring the consistency of the whole..

9.5.1.2 Inputs

The following inputs should be available for the 'perform system structure design' process:

a) functional break down and functional flows;

b) potential dysfunctions;

b) function to operational activity traceability;

c) function allocation to high-level System elements;

d) system requirements, measure of performance (MOP), verification means and requirement traceability.

9.5.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the ‘perform system structure design’ process.

a) System decomposition (system breakdown structure -SBS) is defined.

b) System structures and interfaces of the logical or physical system elements are established.

c) Alternatives are evaluated and best solutions are selected.

d) Key performance attributes for the logical system elements are defined or derived from the upper level.

e) Models are validated using execution or simulation.

f) System elements are traced to requirements.

g) Product breakdown structure (PBS), CIs, bill of materials etc. are defined.

h) Design data required for system analysis are available for:

   1) safety analysis;

   2) reliability availability maintainability testability (RAMT);

   3) integrated logistics support (ILS) for service analysis;

   4) integration verification and validation (IV and V) preparation;

   5) human factors engineering (HFE).

9.5.1.4 Tasks

The organization shall implement the following tasks with respect to the ‘perform system structure design’ process.

— Realize system logical structure.

— Realize system physical structure.

— Realize and manage traceability.

9.5.2 Realize system logical structure

The goal of this task is to decompose the system to produce a logical system breakdown structure, and define the interactions between the system elements. When several candidate structures emerge, trade-offs are realized to identify best logical structuring.

a) The method should support the goal to realize system logical structure with the following capabilities:

   1) realizing system decomposition (system breakdown structure -SBS);

   2) defining subsystems communication or functional interfaces;

   3) identifying alternatives and realize trade-offs;

4) refining models.

b) Tool capabilities should support the goal to realize system logical structure by allowing the user to do the following, in addition to the common tool capabilities:

1) define the logical architecture of the System, which creates the system elements of the next system layer;

2) define the non-functional requirements for the system elements of the created logical architecture;

3) define the key performance attributes or characteristics for the logical system elements. The performance attributes or characteristics are derived from the architecture of the level above;

4) validate the model by execution or simulation;

5) trace the system elements of the created logical architecture to its requirements.

9.5.3 Realize system physical structure

The goal of this task is to identify the physical or technological elements which implement the logical system structure. When several candidate structures emerge, trade-offs are realized to identify best physical elements as well as to take make or buy decisions.

a) The method should support the goal to realize system logical structure with the following capabilities:

1) defining physical structures implementing logical systems and system elements;

2) identifying alternatives and realize trade-offs;

3) defining product breakdown structure (PBS) and produce CIs, bill of materials etc.;

4) providing the models to HFE for HFE analysis, if required;

5) refining models.

b) Tool capabilities should support the goal to realize system physical structure by allowing the user to do the following, in addition to the common tool capabilities:

1) create the product breakdown structure (PBS);

2) provide the design data to safety for safety analysis;

3) provide the design data for reliability availability maintainability testability (RAMT);

4) provide the design data for integrated logistics support (ILS) analysis;

5) provide the design data for virtual engineering;

6) provide the design data for integration verification and validation (IV and V) preparation;

7) provide the results for HFE.

9.5.4 Realize and manage traceability

The goal of this task is to realize and manage traceability between logical and physical model elements as well as the traceability with the domain design elements.

a) The method should support the goal to realize and manage traceability with the following capabilities:

1) tracing logical system elements to physical elements;

2) tracing, making or buying component with the corresponding requirements.

b) Tool capabilities should support the goal to realize and manage traceability by allowing the user to do the following:

1) support and document traceability.

9.6 Perform system analysis

9.6.1 Principal constituents

9.6.1.1 Purpose

The purpose of this process is system analysis as safety, resilience, reliability and security analysis. It also considers simulation, which helps analyse the system functions or dysfunctions during design and operational phases and to gain insights.

9.6.1.2 Inputs

The following inputs should be available to perform the 'perform system analysis' process:

a) modelling strategies;

b) system models;

c) criteria for analysis and decisions.

9.6.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the 'perform system analysis' process.

a) The system model is analysed from a safety, resilience, reliability, security perspective and other quality attributes ("ilities").

b) The levels of risks are identified.

c) The safety, reliability, resilience and security levels are defined.

9.6.1.4 Tasks

The organization shall implement the following tasks with respect to the 'perform system analysis' process.

— Perform safety or reliability analysis.

— Perform security analysis.

— Perform resilience analysis.

9.6.2 Perform safety or reliability analysis

The goal of this task is to identify and analyse the risk of hazards in the system, and to determine whether the system can perform its required functions under stated conditions for a specified period of time.

a) The method should support the safety or reliability analysis goal with the following capabilities:

1) determining required reliability (or reliability targets) and safety margins;

2) capturing or creating safety and reliability information in the system model;

3) reasoning on and analysis of this safety and reliability information:

   i) directly on the model;

   ii) indirectly via model transformations to transfer data to and from external tools;

4) visualizing this safety and reliability information;

5) verifying the correctness, consistency and completeness of this safety and reliability information;

6) tracing safety or reliability or system information within the safety or reliability information itself as well as to related model elements (e.g. requirements, design elements, parametric models, test cases, and test results).

b) Tool capabilities should support the safety and reliability analysis goals by allowing the user to do the following:

1) support the fault tree analysis (FTA) and failure mode and effects analysis (FMEA) analysis techniques;

2) estimate the reliability.

NOTE The MBSA approach can be used, as an enabler, for safety.

9.6.3 Perform security analysis

The goal of this task is to help ensure the prevention, detection, and response to unintended behaviour caused by outside influences. Such unintended behaviour includes theft, causation, prevention, or modification of system behaviour, or interruption in the delivery of system services.

a) The method should support the security analysis goal by performing a threat analysis with the following capabilities:

1) identifying trust boundaries;

2) identifying assets to be protected;

3) assigning values to the assets to decide the level of cost and effort required to protect them;

4) for each asset, identifying the vulnerabilities: means by which the asset can be compromised;

5) identifying the threats (attack vectors), that is, the means by which the vulnerability can be exploited;

6) identifying the security levels;

7) for each threat, identifying a countermeasure to prevent the attack.

b) Tool capabilities should support security analysis goals by allowing the user to do the following:

1) identify the trust boundaries;

2) identify the assets;

3) describe the threats;

4) set the security levels;

5) identify the countermeasure to prevent the attack.

9.6.4 Perform resilience analysis

The goal of this task is to help ensure that the system can deliver required capability when faced by adversity. The analysis addresses the system's ability to avoid, withstand, and recover from potential adversities.

a) The method should support the resilience analysis goal by performing threat and vulnerability analyses with the following capabilities:

   1) identifying critical capabilities and the level to which they are required;

   2) identifying potential adversities and vulnerabilities;

   3) assessing the capability delivered in the face of adversity and gauge it against that which is required.

b) Tool capabilities should support the resilience analysis goal by allowing the user to do the following:

   1) identify system vulnerabilities;

   2) identify required system capabilities;

   3) identify potential adversities;

   4) set required capability levels;

   5) specify resilience metrics and targets;

   6) identify countermeasures for achieving resilience; to include methods of avoiding, withstanding and recovering from adversity.

9.7 Perform domain design integration

9.7.1 Principal constituents

9.7.1.1 Purpose

The purpose of this process is to build system and software design models, analysis models and to help ensure the higher-level integration of the design models with each other and traceability to the higher-level system models.

It defines how the system requirements are achieved through the combination of elements (software subsystems, physical products or operational enablers) to provide the required system features. Depending on the level reached, the specification of those features may lead either to engineering new subsystems, to develop specific software or hardware items, or to reuse existing components (e.g. commercial-off-the-shelf (COTS) or non-development items (NDIs)). During the definition and analysis of design, the features of the elements or items are be defined and thus through an understanding of their purpose the element or item requirements are identified. Various methods of analysis are applied during this process that take into account the planned performance and qualities perspectives necessary to help ensure a fit for purpose and fit for life solution. Key to the success of these activities is –the focus on human factors and organizational factors – and the inclusion of speciality and discipline experts such as those concerned with software, physical, operations, safety, security, commercial etc. For example, in a physical domain, experts can contribute by providing:

— identification of mechanical, thermal, propulsion and electrical structures and features;

— information on technologies or materials maturity, existing products capabilities;

— analyses in terms of mass, volume, physical allocation;

— analysis and demonstration of requirements or constraints compliance, analyses in terms of performance; all these elements are used from trade-off analyses until the design consolidation.

More generally, experts of different domains can contribute relevant information using several types of discipline-specific models including safety models, mechanical, thermal, or electrical models, or UML models specifying software and data models:

— mechanical design;

— electrical design;

— electronic design;

— control design;

— software design;

— information on technologies or materials maturity, existing products capabilities;

— analysis and allocation of mass, volume, power consumption, and central processing unit (CPU) budgets;

— thermal analysis;

— analysis of the overall achievable performance;

— analysis and demonstration of requirements or constraints compliance.

All this information is provided in conjunction with the trade-off activities until design consolidation.

9.7.1.2 Inputs

The following inputs should be available to perform the 'perform system and software design and higher-level integration' process:

a) modelling strategies;

b) criteria for analysis and decisions.

9.7.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the 'perform system and software design and higher-level integration' process.

a) System elements design models are defined.

b) Models are integrated and analysed according to the specialty engineering analysis results.

c) Interfaces between system and software design models are defined or refined.

d) Traceability of the system and software design models to the architectural models and to the higher-level system models is established.

9.7.1.4 Tasks

The organization shall implement the following tasks with respect to the 'perform system and software design and higher-level integration' process.

— Perform system design modelling.

— Support system integration with the use of models.

9.7.2 Perform system design modelling

The goal of this task is to provide sufficient detailed models of the system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

a) The method should support the goal to perform system design modelling with the following capabilities:

   1) allocating the system requirements to system elements;

   2) providing, specifying the V and V with inputs;

   3) transforming architecture models to design models or realize models for system elements;

   4) modelling variants and performing trade-offs;

   5) refining or defining the interfaces between the system element models and with external models;

   6) developing the technical architecture;

   7) validating the technical design e.g. by analysis, simulation, prototype, or mock-up;

   8) defining the interface control document (ICD) between the elements of the technical architecture, if required; typically the ICDs are developed at line replaceable unit (LRU) level;

   9) handing over the design elements to the different domains.

b) Tool capabilities should support the goal to perform system design modelling by allowing the user to do the following:

   1) support for component design modelling (mechanical, thermal, electrical, software, etc.);

   2) support for specific modelling and simulation languages;

   3) validate the model by execution or simulation;

   4) provide support for the requirement allocation and traceability;

   5) provide the design data to analysis tools (safety, RAMT, ILS, virtual testing; IV and V);

   6) provide the results to HFE for HFE Analysis;

   7) derive the optimal technical architecture for the given constraints;

   8) define the non-functional requirements or constraints for the physical architecture.

9.7.3 Support system integration with the use of models

The goal of system integration is to synthesize a set of system elements into a realized system (product or service) that satisfies system requirements, architecture, and design. “System” in this context is anything that is being constructed from its elements into an integrated whole. The perform integration processes concerns the integration plan, which defines the resources (environments, people, and specialist tools), targets, tests, procedures etc. for integration and the provision of the integration report, which contains the results, issues, risks, encountered. There are several possible integration approaches and techniques. Any of these may be used individually or in combination. As a general rule of thumb, however, the earlier you can perform integration activities to verify design the better the outcome is likely to be. Integration of the system has to be completed (at least to an acceptable level) prior to verification readiness review.

Integration is supported in MBSSE by designing system models or simulations to inter-operate with other system models or simulations at each system model level (i.e. typical levels include strategic context, operational behaviour, functional behaviour, functional or logical structure and physical structure each level representing capabilities, operational needs, system needs, and solution definition); for example, for the purpose of increased performance, cost benefit, or synergism. Interfaces are defined so that simulations can interact in the same way the system is intended to operate. Multiple benefits or savings can be gained from increased synergism and use over time and across activities. Integration is achieved through reuse or upgrade of legacy programs used by the system, or of the proactive planning of integrated development of new simulations. In this case, integration is accomplished through the planned utilization of models, simulations, or data for multiple iterations or applications over the system life cycle.

Virtual system integration is totally dependent on models. A requirements level behavioural model is created for each subsystem of the system to be integrated. These subsystem behaviour models are integrated to produce an overall system model. Then, the subsystem requirements level behaviour models are replaced progressively with solution level behaviour models, maintaining the integration. Solution level behaviour models are replaced progressively with prototype subsystems, and subsequently fully engineered and verified subsystems, maintaining integration through the whole evolution from totally a model to totally physical.

The goal of this task is to support system integration with the use of models.

a) The method should support system integration with the use of model with the following capabilities:

   1) designing the models or simulation that have to inter-operate;

   2) integrating or synthesizing the set of system elements into a realized system (product or service) using the appropriate data;

   3) confirming that the models satisfy system requirements, architecture, and design.

b) Tool capabilities should support system integration with the use of model by allowing the user to do the following:

   1) provide a system integration report which contains the results, issues, and risks;

   2) provide a report, which contains simulation, system models and data used for systems integrations over the system life cycle.

9.8 Perform system verification and validation

9.8.1 Principal constituents

9.8.1.1 Purpose

The purpose of this process is to define activities for demonstrating that during the system development life cycle and based on models, the system or system element fulfils its requirements and characteristics, meets the needs expressed by stakeholders, and fulfils its intended purpose in its intended operational environment.

NOTE 1 Similar to the system verification and validation, the model-based early verification process determines that the “model is built right”. The model-based early validation process determines that the “right model is built”.

NOTE 2 Verification and validation techniques, includes peer reviews, inspections, demonstrations, testing, model checking, runtime verification, fault diagnosis, analysis, simulations, formal proof and more exploratory techniques such as the use of theorem proving. The objectives of each technique are complementary.

NOTE 3 Verification and validation plans can be generated from the digitalized SE data. These plans define the resources (environments, people, and specialist tools), verification procedures (test titles, purposes and mechanisms) and the verification reports (contains the results, issues, risks encountered).

9.8.1.2 Inputs

The following inputs should be available to perform the 'perform system verification and validation' process:

a) system models and discipline-specific models including System-level use cases, scenarios;

b) criteria for analysis and decisions;

c) system requirements, MOP, verification plans and means, requirement traceability;

d) stakeholder requirements, MOE, validation plans and means.

9.8.1.3 Outcomes

The following outcomes shall be achieved as a result of the successful implementation of the 'perform system verification and validation' process.

a) The test cases and associated verification procedures are developed.

b) The system model is verified to help ensure that it fulfils system requirements.

c) The system model is validated to help ensure that it meets the stakeholder needs and the business or mission needs.

9.8.1.4 Tasks

The organization shall implement the following tasks with respect to the 'perform system verification and validation' process.

— Prepare model-based verification and validation.

— Perform model-based verification and validation.

— Manage results.

9.8.2 Prepare model-based verification and validation

The goal of this task is to prepare the verification and validation plans and environment.

a) The method should support the goal to prepare model-based verification and validation with the following capabilities:

   1) generating the verification and validation plans;

   2) setting up required tool environment;

   3) collecting and digitalizing required data;

   4) preparing models and mock-ups required for V and V: The verification system can be modelled using the same activities and artefacts described above for modelling the operational system;

   5) updating requirements management database to trace the system requirements and design information to the system verification methods, test cases, and results.

b) Tool capabilities should support the goal to prepare model-based verification and validation by allowing the user to do the following, in addition to the common tool capabilities:

   1) implement formal modelling;

   2) generate the system verification plans for the system and the system elements;

   3) realize user interface mock-ups to support demonstrations or launching simulations;

4) implement different techniques such as model checking, runtime verification, fault diagnosis, formal proof.

9.8.3 Perform model-based verification and validation

The goal of this task is to verify and validate the system using models during early stages of the system life cycle.

a) The method should support the goal to perform model-based verification and validation with the following capabilities:

   1) executing and simulating models within actual or simulated operational environment conditions;

   2) performing the defined V and V techniques according to the plans and using verified and validated models.

b) Tool capabilities should support the goal to perform model-based verification and validation by allowing the user to do the following:

   1) support in model execution and simulation to analyse.

9.8.4 Manage results

The goal of this task is to manage the results.

a) The method should support the goal to manage results with the following capabilities:

   1) recording verification results and anomalies encountered;

   2) generating reports;

   3) managing verified model baseline.

b) Tool capabilities should support the goal to manage results by allowing the user to do the following:

   1) store results;

   2) document generation capability;

   3) connect with CM tool.

Annex A
(informative)

Instantiation and customization of an MBSSE reference framework

A.1 Abstractness and generalizability of an MBSSE reference framework

A framework is a logical structure for classifying and organizing complex information. In any given domain, a reference framework provides a common backplane for consistency, collaboration, sharing, and reuse. With an appropriate reference framework, the work of individual projects, programs, divisions and partners will be coordinated with just enough formality to ensure that the many moving parts can fit together when and where needed.

Although MBSSE is not a scientific paradigm shift of systems and software engineering, it is really a brand-new working mode for them which lays the cornerstone of digital transformation and digital engineering for an organization. The adoption of MBSSE will greatly facilitate the execution of the life cycle processes described in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 due to the models as the new working media and the models' evolution as the single source of truth and major source of knowledge about the System-of-interest and its life cycle processes. However, the process reference models defined in these two standards do not provide enough systematic supports to MBSSE. The descriptions of information management process in the process group of technical management processes and knowledge management process in the process group of organizational project-enabling processes are too general. Furthermore, the definition of the concept "life cycle model" in ISO/IEC/IEEE 24748-1 only addresses two dimensions of the system-of-interest, which are the evolution stages of the system-of-interest and a framework of processes and activities, and are lack of the emphasis of the dimension of the models evolution and organization cognition. Therefore, a multi-dimension reference framework is needed first of all to support MBSSE.

Based on the definition of "life cycle model" and the essential characteristics of MBSSE approach and working mode, a three-dimensional MBSSE reference framework (see Figure A.1) is proposed to organize and formalize the execution of the life cycle processes described in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 and support the digital transformation of systems and software engineering. Among all reference frameworks, the MBSSE reference framework has the highest abstractness and generalizability because its three dimensions and their twelve aspects (the ordinate names of each dimension) address the nature of human-being transforming the world and problem-solving process.

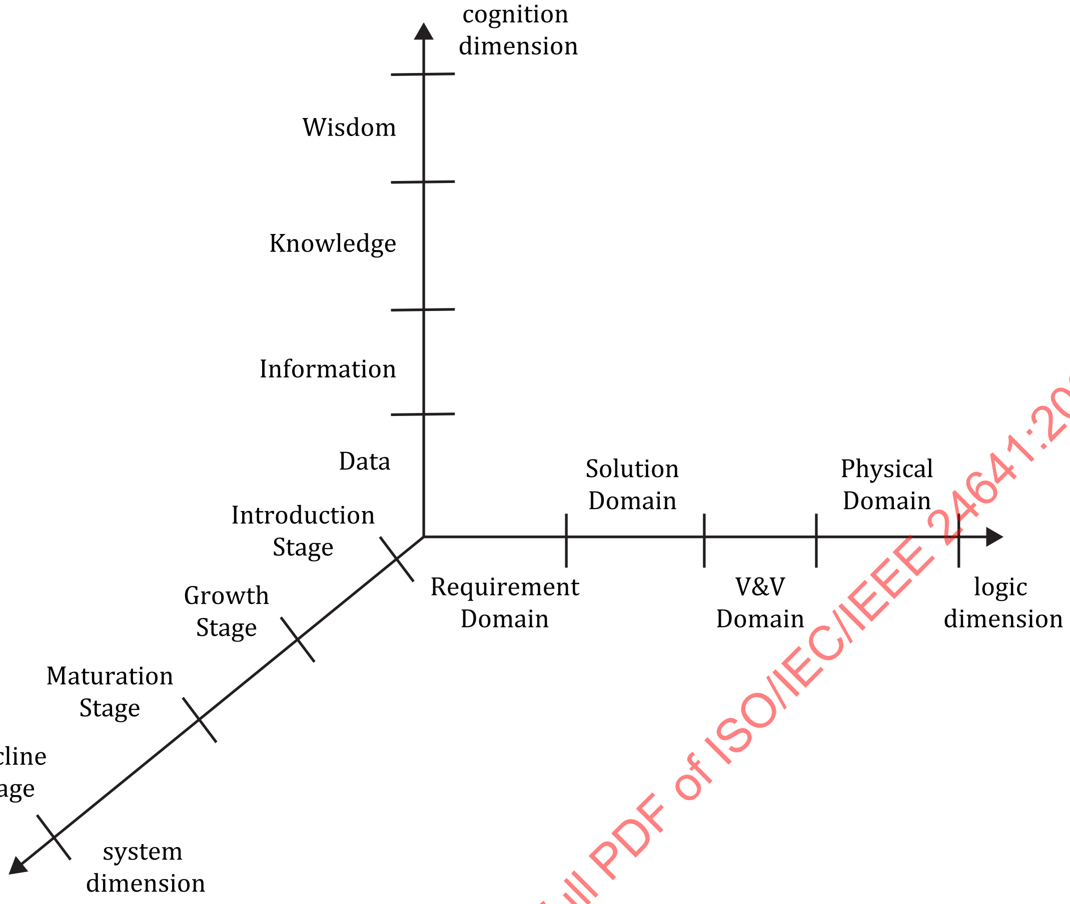

Figure A.1 — MBSSE reference framework

According to its definition in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 24748-1, the life cycle model of human-made systems and entities has two attributes, (1) evolutionary stages, and (2) framework of processes and activities. The two orthogonal dimensions of MBSSE reference framework, system dimension and logic dimension, are representation of these two attributes of life cycle models respectively.

The system dimension emphasizes the interactions among the stakeholders along the life cycle stages of the system-of-interest. It is a dimension about human-made systems and entities. The business meaning of the arrow of this dimension is to increase the readiness level of the system-of-interest as time goes on.

The logic dimension emphasizes the performing of system engineering processes on the system-of-interest. It is a dimension about activity or event. The business meaning of the arrow of this dimension is to promote the capability maturity of systems engineering processes in a project team as time goes on.

The cognition dimension records the processes and results of the understanding and transformation of the objective world. It is a dimension about individuals and organization. The business meaning of the arrow of this dimension is to promote the competence maturity of an organization and individuals as time goes on. On the one hand, MBSSE dramatically enhances the effectiveness and efficiency of the transformation and collaboration among the DIKW pyramid; on the other hand, the cognition dimension provides a framework of information and knowledge management of MBSSE.